UNIVERSITY OF COPENHAGEN

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

Accountable and Explainable Methods for Complex Reasoning over Text

Pepa Kostadinova Atanasova

Supervised by Isabelle Augenstein, Jakob Grue Simonsen and Christina Lioma

Submission date: 16th September 2022

Acknowledgements

This Ph.D. thesis is the ultimate outcome of the challenging yet rewarding Ph.D. journey, which would not have been possible without the support of many people. First and foremost, I would like to express my deepest appreciation to my supervisors, Isabelle, Jakob, and Christina. They have provided me with invaluable guidance, kindhearted support, much-needed critical advice, and the freedom to pursue my research interests. To my principal supervisor, Isabelle, I am grateful for being an inspiring role model, for establishing a stimulating lab environment, and for providing opportunities for professional growth.

I also sincerely thank the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199 that has funded my Ph.D. research.

Furthermore, I could not have undertaken this journey without the initial guidance from my Master's degree supervisor, Preslav. I continue to be indebted to his support and introduction to research and academia. I am also grateful to all the bright and kindhearted lab mates from the CopeNLU group for filling my journey with many memorable moments and engaging discussions.

Next, I would like to express my gratitude to all the remarkable collaborators I have been fortunate to work with. They have shared research interests with me and provided valuable perspectives and contributions to my line of research. I was also fortunate to conduct two research internships at Google and Meta, where Abe, Alyssa, Cong, Yashar, and Wenhan provided outstanding mentorship and made a lot of efforts to create an inclusive and productive environment despite the conditions during the pandemic.

On a personal note, this endeavor would not have been possible without the unfailing support of my family and friends. To Ludmila and Kostadin, I am forever indebted for their unconditional and loving encouragement that kept me motivated and confident. To Atanas, I am forever grateful for the enthusiasm to make a new home in a different country and for always being a continuous source of sanity, joy, and encouragement.

Abstract

A major concern of Machine Learning (ML) models is their opacity. They are deployed in an increasing number of applications where they often operate as black boxes that do not provide explanations for their predictions. Among others, the potential harms associated with the lack of understanding of the models' rationales include privacy violations, adversarial manipulations, and unfair discrimination. As a result, the accountability and transparency of ML models have been posed as critical desiderata by works in policy and law, philosophy, and computer science.

In computer science, the decision-making process of ML models has been studied by developing accountability and transparency methods. Accountability methods, such as adversarial attacks and diagnostic datasets, expose vulnerabilities of ML models that could lead to malicious manipulations or systematic faults in their predictions. Transparency methods explain the rationales behind models' predictions gaining the trust of relevant stakeholders and potentially uncovering mistakes and unfairness in models' decisions. To this end, transparency methods have to meet accountability requirements as well, e.g., being robust and faithful to the underlying rationales of a model.

This thesis presents my research that expands our collective knowledge in the areas of accountability and transparency of ML models developed for complex reasoning tasks over text. First, this thesis contributes with two methods for accountable ML models. They generate adversarial inputs and a diagnostic dataset demonstrating significant model vulnerabilities and suggesting ways to correct those. In the area of transparency of ML models, this thesis advances the state-of-the-art with methods generating textual explanations that are further improved to be fluent, easy to read, and to contain logically connected multi-chain arguments. Finally, this thesis makes contributions in the area of diagnostics for explainability approaches with a set of properties for evaluating existing explainability techniques and methods for enhancing those further in produced explanations. All of the contributions are empirically tested on complex reasoning tasks over text, including fact checking, question answering, and natural language inference.

Resumé

En væsentlig kilde til bekymring i forskning om maskinlæringsmodeller er modellernes uigennemskuelighed. Sådanne modeller benyttes i stigende grad til anvendelser, hvor de opererer som “black boxes” som ikke forklarer deres forudsigelser eller beslutninger. Blandt de potentielle farer knyttet til mangel på forståelse for modellernes rationaler er krænkelser af privatlivet, fjendtlig manipulation og uretfærdig forskelsbehandling. Som følge heraf er ansvarlighed (en. *accountability*) og gennemskuelighed (en. *transparency*) for maskinlæringsmodeller blevet foreslået som kritisk vigtige designkriterier af forskning i politik, jura, filosofi og datalogi.

I datalogi er maskinlæringsmodellers beslutningsprocesser blevet undersøgt ved udvikling af metoder for ansvarlighed og gennemskuelighed. Flere metoder for undersøgelse af ansvarlighed, herunder brugen af fjendtlige angreb og diagnostiske datasæt, har påvist eksistensen af sårbarhed for modeller, og at sådanne sårbarheder kan føre til ondsindet manipulation af resultater, eller systematiske fejl i modellernes forudsigelser. For at imødegå dette, er det nødvendigt, at metoder til at sikre gennemskuelighed tillige opfylder en række krav for ansvarlighed, f.eks. at udvise robusthed og være tro mod de underliggende rationaler i modellerne.

Denne afhandling fremlægger min forskning, som udvider den samlede viden inden for ansvarlighed og gennemskuelighed for maskinlæringsmodeller, som er udviklet til at løse opgaver, der involverer komplekst ræsonnement om tekstdata. For det første bidrager afhandlingen med to ny metoder for ansvarlige maskinlæringsmodeller, herunder skabelsen fjendtlige input og et diagnostisk datasæt, som påviser væsentlige sårbarheder i modeller, og foreslår metoder til at afhjælpe disse sårbarheder. Inden for gennemskuelighed af maskinlæringsmodeller, bidrager afhandlingen med metoder til automatisk skabelse af forklaringer på skriftform, som yderligere forbedres til at benytte flydende sprog, er letlæselige og indeholder logisk sammenhængende argumentationskæder. Sluttelig bidrager afhandlingen til diagnostik af metoder til forklarlighed af maskinlæringsmodellers forudsigelser ved at definere en række egenskaber med hvilke allerede eksisterende forklarlighedsmetoder - og metoder til forbedring af sådanne - kan evalueres. All afhandlingens bidrag er eksperimentelt afprøvet på

problemer, som involverer komplekst ræsonnement om tekstdata, herunder faktatjek, automatisk besvarelse af spørgsmål, og følgeslutninger i naturligt sprog.

Publications

This thesis includes the following papers as chapters, listed in the order of their appearance (* denotes equal contribution):

1. (Atanasova et al., 2022) Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Fact Checking with Insufficient Evidence. 2022. Transactions of the Association for Computational Linguistics, pages 746–763.

2. (Atanasova et al., 2020c) Pepa Atanasova*, Dustin Wright*, and Isabelle Augenstein. Generating Label Cohesive and Well-Formed Adversarial Claims. 2020. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3168–3177, Online. Association for Computational Linguistics.

3. (Atanasova et al., 2020b) Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating Fact Checking Explanations. 2020. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7352–7364, Online. Association for Computational Linguistics.

4. (Jolly et al., 2021) Shailza Jolly, Pepa Atanasova, and Isabelle Augenstein. Generating Fluent Fact Checking Explanations with Unsupervised Post-Editing. 2021. Preprint arXiv:2112.06924 .

5. (Ostrowski et al., 2021) Ostrowski, Wojciech, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. Multi-hop fact checking of political claims. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, pages 3892-3898, Montreal, Canada.

6. (Atanasova et al., 2020a) Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A Diagnostic Study of Explainability Techniques for Text Classification. 2020. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3256–3274, Online. Association for Computational Linguistics.

7. (Atanasova et al., 2021) Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Diagnostics-Guided Explanation Generation. 2022. Proceedings of the AAAI Conference on Artificial Intelligence 36 (10), pages 10445-10453.

The following list of papers were also published during my Ph.D. studies. As they are unrelated to the topic of the thesis, they are not included in it. The topics of these publications include: dealing with limited labelled data (1), offensive language identification (2-3), fact checking and check-worthiness detection (4-6), dialogue system evaluation (7).

1. (De Bruyne et al., 2022) Luna De Bruyne, Pepa Atanasova, and Isabelle Augenstein. Joint emotion label space modeling for affect lexica. 2022. Computer Speech & Language 71, pages: 101257.

2. (Zampieri et al., 2020) Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). 2020. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pages 1425–1447, Barcelona (online).

3. (Rosenthal et al., 2021) Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. 2021. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 915–928, Online. Association for Computational Linguistics.

4. (Atanasova et al., 2019b) Atanasova, Pepa, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness. 2019. International Conference of the Cross-Language Evaluation Forum for European Languages. Springer, Cham.

5. (Vasileva et al., 2019) Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. It Takes Nine to Smell a Rat: Neural Multi-Task Learning for Check-Worthiness Prediction. 2019. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pages 1229–1239, Varna, Bulgaria.

6. (Atanasova et al., 2019c) Pepa Atanasova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Georgi Karadzhov, Tsvetomila Mihaylova, Mitra Mohtarami, and James Glass. Automatic Fact-Checking Using Context and Discourse Information. 2019. J. Data and Information Quality 11, 3, Article 12 (September 2019), 27 pages.

7. (Atanasova et al., 2019a) Pepa Atanasova, Georgi Karadzhov, Yasen Kiprov, Preslav Nakov, and Fabrizio Sebastiani. Evaluating Variable-Length Multiple-Option Lists in Chatbots and Mobile Search. 2019. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19). Association for Computing Machinery, New York, NY, USA, pages: 997–1000.

Contents

List of Figures

List of Tables

Part I

Executive Summary

Executive Summary

1

1.1 Introduction

Recent progress in the field of machine learning and specifically in natural language processing has been driven by the development of large models pre-trained on massive amounts of data (Vaswani et al., 2017; Devlin et al., 2019). Notably, such models have been used to extend the state of the art in a broad range of tasks (Wang et al., 2018) and have been deployed in an increasing number of downstream applications (Angwin et al., 2022; Obermeyer et al., 2019; Barocas and Selbst, 2016; Lambrecht and Tucker, 2019). On the other hand, the models' increased architectural complexity has raised concerns about the decreased ability of humans to understand the opaque decision-making processes of these models (Raji et al., 2020; Bender et al., 2021). To this end, methods for accountable and transparent machine learning models have been developed that verify and unveil the reasons behind the models' predictions (Raji et al., 2020). These methods assess critical aspects of machine learning models beyond the achieved task performance, such as vulnerability to adversarial decision manipulations (Kreps et al., 2022; Agarwal et al., 2019), unfair embedded biases towards certain groups and individuals (Kiritchenko and Mohammad, 2018; Raji et al., 2020; Ntoutsi et al., 2020), privacy violations (Carlini et al., 2019), and generalisation to out-of-distribution samples (Koh et al., 2021). Accountability and transparency methods can further be used as means to engender trust in a model's decisions (Ribeiro et al., 2016b), expand the knowledge about a downstream task (Forde et al., 2022; Ghandeharioun et al., 2022) and debug and improve a model's decision-making process (Anders et al., 2022; Abid et al., 2022).

This section introduces accountability and transparency methods for machine learning models from the perspective of computer science and, in particular, for complex reasoning tasks over text, such as fact checking, question answering, and natural language inference. The papers included in the following chapters of this Ph.D. thesis are cross-referenced where relevant. Section 1.2 provides a detailed overview of the contributions of the separate publications included in this Ph.D. thesis in the areas of accountable and transparent machine learning models. Section 1.3 offers an introspective summary of the contributions and suggests prospects for future work.

1.1.1 Accountability

The accountability of a machine learning model is verified by methods that analyse the model's outputs on specifically crafted instances in order to detect and correct for flaws in its reasoning process, such as reliance on spurious correlations. To this end, accountability methods usually produce datasets used to inspect whether the model's outputs are the desired outcomes for the instances in the created dataset. The produced datasets can be challenging (Nie et al., 2020; Atanasova et al., 2022) or adversarial in nature (Atanasova et al., 2020c; Song et al., 2021). They can reveal specific model vulnerabilities such as a model's reliance on spurious features (McCoy et al., 2019; Schuster et al., 2019), vulnerability to maliciously manipulated inputs (Atanasova et al., 2020c; Song et al., 2021; Guo et al., 2021), lack of generalisation to out-of-distribution samples (Koh et al., 2021), and specific reasoning skills that a model failed to acquire (Dua et al., 2019; Talmor et al., 2020). Moreover, such datasets can steer the development of model architectures designed to handle the revealed model vulnerabilities (Zhao et al., 2020). They can also provide additional training data points to enhance the performance of existing models on the challenges presented by these datasets (Nie et al., 2020; Schuster et al., 2021).

1.1.1.1 Diagnostic Challenge Datasets

Owing to improvements in computational power and the development of effective machine learning models, such as models with the Transformer architecture (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019) and pre-trained language models (Howard and Ruder, 2018; Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020), the time for achieving near-human performance on new tasks has decreased to a few months (Kiela et al., 2021). Existing work, however, has questioned whether a model's performance on a dataset indicates it has learned meaningful features required to solve the task underlying the dataset (Bowman and Dahl, 2021). Studies have found that, on the contrary, machine learning models often learn to rely on spurious correlations located in the training data (McCoy et al., 2019; Ribeiro et al., 2020). These findings have motivated the research on *challenge datasets* that diagnose whether models learn specific meaningful features and do not overfit to correlations in the training set. Moreover, Dua et al. (2019); Nie et al. (2020); Kiela et al. (2021) propose dynamic benchmarks where challenge datasets are collected via an iterative human-and-model-in-the-loop procedure. In the first step of the procedure, human annotators collect a challenge dataset for which a given model cannot predict the correct labels. In the second step, the training split of the challenge dataset is used to train a better-performing model. The two steps can be applied repeatedly, creating a moving target, rather than a static benchmark that models quickly overfit to.

Predominantly, challenge datasets are designed around particular sets of reasoning skills such as logical reasoning (Liu et al., 2020a), linguistic capabilities (Ribeiro et al., 2020; Saha et al., 2020), and common-sense inference (Zellers et al., 2018) (see Section 1.1.4 for an overview of reasoning skills). There also exist challenge datasets that test a model's ability to detect when the provided input is insufficient to make the correct decision (Rajpurkar et al., 2018; Atanasova et al., 2022). Other challenge datasets are contrastive in nature and verify whether a model can detect minimal changes in the input that lead to a change in the prediction (Gardner et al., 2020; Kaushik et al., 2020; Sen et al., 2021).

Challenge datasets can further be categorised as model-agnostic (Talmor et al., 2020; Saha et al., 2020; Schuster et al., 2019) or created with a model in the loop (Nie et al., 2020). Model-agnostic datasets are usually produced manually, where expert knowledge is utilised to construct tests for different skills. There is, however, no guarantee that the resulting dataset will cover potential model flaws. On the other hand, challenge datasets created with a model in the loop could cover deficiencies only of the employed models. The latter can be produced manually, where data creators are incentivised to deceive a given model (Wallace et al., 2019b; Nie et al., 2020) or with automated data generation techniques (Atanasova et al., 2022; Le Bras et al., 2020).

This thesis presents advances in the area of challenge datasets for model accountability with novel methodology, insights, and models' accountability improvements. *Paper 1* (§2) presents a novel automated method for constructing a contrastive model-in-the-loop challenge dataset to study what information models consider sufficient for producing a prediction. In knowledge-intensive tasks such as fact checking, it is markedly crucial to make predictions only when the presented input information is sufficient and otherwise indicate it is not enough. The technique introduced in this publication employs three separate models in the loop to preserve validity beyond a single model. The latter also allows one to compare and contrast different models' deficiencies. The introduced method is further used to improve models' performance on instances with insufficient evidence information.

Paper 5 (§6) presents a challenge dataset for multi-hop reasoning for the task of fact checking. The dataset contains real-world claims with manual annotations of sets of logically connected evidence pieces that lead to the final verdict of a claim. It enables accountability investigations of whether a model employs multi-hop reasoning by logically connecting evidence chunks needed for a prediction as opposed to predicting based on a single inference step. Based on the challenge dataset, the publication presents findings on the multi-hop reasoning capabilities of two existing models where an architecture designed specifically to conduct multi-hop reasoning performs the best.

1.1.1.2 Adversarial Attacks

Adversarial attacks reveal model vulnerabilities to changes in the input that manipulate the model to produce a target prediction, different from the correct one. Adversarial attacks can be performed at training or test time. Training-time adversarial attacks (Qi et al., 2021; Kurita et al., 2020; Wallace et al., 2021) manipulate either the model weights or the training data, assuming unrestricted access to the training process of a model. Test-time adversarial attacks unveil model vulnerabilities of already trained models and can assume access to the parameters of a model – white-box attacks (Atanasova et al., 2020c; Guo et al., 2021), or no access to them – black-box attacks (Chen et al., 2021; Berger et al., 2021).

Adversarial attacks usually perform manipulations of the input, which are the smallest possible changes required to achieve a target prediction. For textual inputs, changes can be performed at character (Eger et al., 2019), word (Mozes et al., 2021; Zang et al., 2020), or sentence level (Jia and Liang, 2017; Iyyer et al., 2018). Due to the discrete nature of textual inputs, performing input manipulations is additionally challenging as there is a reduced number of possible manipulations that preserve the validity of the input. In contrast, in tasks where the input spaces are continuous, e.g., image and time-series analysis, it is possible to perform numerous input perturbations that do not harm the overall realistic outlook of the input (Papernot et al., 2016; Goodfellow et al., 2014; Szegedy et al., 2013) and are sometimes even invisible to the human eye (Szegedy et al., 2013; Moosavi-Dezfooli et al., 2016).

The potency of adversarial attacks is commonly measured as the number of samples where a model's prediction can be manipulated. One example of a potent adversarial technique is the universal adversarial attack, which degrades the performance of a model by inserting a particular textual sequence, termed as a trigger, in all instances (Wallace et al., 2019a; Song et al., 2021). For a more detailed overview of adversarial attacks, refer to Xu et al. (2020); Chakraborty et al. (2021).

In *Paper 2* (§3) of this thesis, I present a novel method for generating test-time white-box adversarial attacks. It draws on the universal adversarial attack approach, which suffers from two precluding deficiencies when applied for inference tasks such as fact checking. First, universal adversarial attacks generate triggers that often invert the meaning of the instances they are inserted in, thus changing also their gold standard label. The method proposed in the paper mitigates this with a novel extension to the universal adversarial attack approach that generates triggers preserving the label of the original instance. Second, universal adversarial attacks produce semantically invalid inputs, as they simply concatenate triggers to existing samples. The method proposed in the paper alleviates this with a conditional language model trained to generate semantically valid statements, which include the found universal triggers. The method is empirically tested with a model for the task of fact checking, where the generated adversarial claims are highly effective in fooling the model and lead to a performance decrease of 23.1 F_1 score points compared to the model's performance on the original claims. At the same time, the generated attacks constitute valid fact checking instances as they have preserved the gold label and the semantic validity of the input.

1.1.2 Explainability

Machine learning models have been heavily criticized for their opaque nature (Zarsky, 2016; Pasquale, 2015). As a result, explanations of their decisions are increasingly needed for debugging, measuring bias and fairness, instilling trust, and making model behavior transparent in general. European law has also introduced a requirement for "the right . . . to obtain an explanation of the decision reached" (Goodman and Flaxman, 2017). Efforts to make models' decisions transparent have led to a growing influx of explainability approaches. What follows next is an overview of common types of explainability techniques. For more details, I refer the reader to Ras et al. (2022); Molnar (2022).

1.1.2.1 Post-hoc Saliency Explanations

Post-hoc explanations reveal the decision process of an already trained model and can be applied to various types of models and different data modalities. Saliency explanations are the most prominent type of post-hoc explanations. They highlight regions of the input according to the region's importance for the prediction of a model. Saliency explanations can be gradient-based, such as the Vanilla Gradient (Simonyan et al., 2013), which computes the gradient of the output w.r.t. the input. Follow-up gradient-based approaches (Kindermans et al., 2016; Springenberg et al., 2014) improve saturation and stability problems of the former. Saliency explanations can also be perturbation-based, e.g., Occlusion (Zeiler and Fergus, 2014), and Shapley Value Sampling (Shapley, 1953), which estimate the contribution of input regions to a model's prediction by occluding regions from the input and observing the corresponding changes in the model's prediction. Finally, simplification-based explanations such as LIME (Ribeiro et al., 2016a) train a simple self-interpretable model to approximate the local decision boundary of the opaque model for each instance.

Contrastive explanations are another type of post-hoc explanations. They find small changes to the input that cause a change in the prediction of a model (Stepin et al., 2021). Studies in social science (Lipton, 1990) argue that contrastive explanations are more intuitive to end users as they unveil the causal factors that explain why an event occurred instead of an alternative event. Contrastive explanations can be constructed by manipulations either at the discrete textual input level or at a latent representation level of a given model. Methods for contrastive explanations at the input level are formulated as search problems where a model, e.g., a language generation model, is trained to apply edit, replace or delete operations at different positions of the input until a change in the prediction of a model is achieved (Wu et al., 2021; Ross et al., 2021). By contrast, Jacovi et al. (2021) observe changes in the predictions of a model caused by projecting its latent representations to a similar representation space where only a particular concept, such as gender, is removed.

1.1.2.2 Natural Language Explanations

Natural language explanations (NLEs) explain model predictions with free text, which contrary to other explanation types, is a natural means of communication that does not require a preliminary clarification phase. Furthermore, NLEs are not constrained to contain only input segments but can also contain generated text that provides more explanation of the model's rationales for the prediction. This gives them greater expressive power in terms of the reasoning they can convey, especially with complex reasoning tasks (see Section 1.1.4) involving rationales beyond what is explicitly stated in the input. Existing datasets with NLEs include the tasks of natural language inference (Camburu et al., 2018; Do et al., 2020), common sense reasoning (Rajani et al., 2019; Zellers et al., 2019), fact checking (Alhindi et al., 2018; Kotonya and Toni, 2020b), and relation extraction (Hancock et al., 2018; Wang* et al., 2020). For a longer discussion of datasets with NLEs, I refer the reader to Wiegreffe and Marasovic (2021).

NLEs are typically produced in a supervised way where models generating NLEs can be trained jointly or separately from the models for the downstream task (Camburu et al., 2018). Prior work has found that training the two tasks jointly leads to more label-informed explanations (Wiegreffe et al., 2021). The explanations can also be conditioned on the predicted label (Hase et al., 2020; Kumar and Talukdar, 2020).

With respect to the input, produced NLEs can be extractive – selected important portions of the input text that constitute a free text explanation, or abstractive – generated text explaining the model's reasoning in words that do not necessarily appear in the input. Extractive NLE techniques still produce natural text, as they usually choose whole sentences from the input (Atanasova et al., 2020b; Thorne et al., 2018). In knowledge-intensive tasks such as question answering and fact checking, producing short extractive explanations from long input documents is often regarded as part of the task, and the performance on the explanation and the downstream task is judged jointly with a unified measure (Thorne et al., 2018; Jiang et al., 2020; Trivedi et al., 2019; Petroni et al., 2021). Compared to abstractive NLEs, extractive NLEs require less training data, which makes them more suitable for low-resource scenarios. Extractive NLEs also produce explanations that are always factual w.r.t. the input, i.e., they contain only information that is correct given the input. The latter cannot be guaranteed for abstractive generated NLEs. On the other hand, abstractive NLEs can be more coherent, provide more information about the model's rationales, and have less redundant information compared to extractive ones (Jolly et al., 2021).

This thesis makes important contributions to the field of explainability establishing methods and datasets for generating NLEs given limited resources and complex reasoning tasks such as fact checking. In *Paper 3* (§4), I introduce the task of generating NLEs for fact checking veracity predictions. Generating veracity explanations is a challenging task, especially when considering real-world claims where training data is limited and claim verification requires constructing fact checking evidence of multiple arguments involving complex reasoning capabilities. In addition, most NLEs for existing tasks contain no more than one sentence per instance. At the same time, the explanations produced for real-world claims have several sentences, which also indicates the complex reasoning required for veracity prediction. Paper 3 proposes a method that generates fact checking explanations in an extractive way and jointly with the task at hand. I find that optimising explanation generation jointly with veracity prediction produces explanations that achieve better coverage and overall quality and are better suited for explaining the correct veracity label than explanations learned solely to mimic human justifications.

As extractive NLEs can lack fluency and coherence and can contain redundant information, *Paper 4* (§5) introduces a method to improve the fluency and readability of fact checking NLEs. The method performs post-editing of extracted NLEs and is the first to explore an iterative unsupervised edit-based algorithm using only phrase-level edits. The proposed method also leads to computationally feasible explanation generation solutions for long text inputs. More importantly, the resulting explanations are found to be fluent, easy to read, and concise.

Finally, *Paper 5* (§6) provides a supervised extractive dataset for producing fact checking explanations that form chains of logically connected arguments. Utilising the dataset, the paper documents the first study on how models construct rationales to verify political claims requiring multi-hop evidence reasoning. The main finding of the study is that the best performance is achieved with an architecture that specifically models multi-hop reasoning over evidence pieces in combination with in-domain transfer learning.

1.1.2.3 Self-Interpretable Models

A simple solution for achieving model transparency is using self-interpretable models that produce predictions in a way that can be interpreted by a non-expert from their inner workings. Examples of self-interpretable models are linear regression (Ge et al., 2018), decision trees (Prentzas et al., 2019), Bayesian models (Letham et al., 2015), and general additive models (Hastie, 2017). These models usually have simple architectures, which struggle to achieve good performance, especially on non-structured input such as text. On the other hand, existing work has attributed self-interpretable capabilities to models using the attention mechanism (Wiegreffe and Pinter, 2019; Meister et al., 2021), which achieve high performance on many tasks. However, the use of attention weights as explanations has also met criticism (Bastings and Filippova, 2020) as they are not always faithful to the prediction rationale of the underlying model (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). One can also often find a different set of attention weights resulting in the same prediction (Serrano and Smith, 2019).

In *Paper 5* (§6) of this thesis, I employ eXtra-hop attention to model the interaction between evidence sentences for fact checking claims. The eXtra-hop attention introduces a way to structure text where the important evidence sentences are linked in a logically connected set of arguments. One of the research questions of the paper explores whether eXtra-hop attention indicates which are the important evidence sentences from the input document for predicting the target task of fact checking. In fact, I find that the model assigns higher eXtra-hop attention weights to evidence sentences employed for the prediction as opposed to the remaining sentences in the input document.

1.1.3 Diagnostic Explainability Methods

The overwhelming influx of explainability approaches increases the number of different explanations that can be produced for a model's prediction (Neely et al., 2021). An open question becomes which explainability approaches produce better-quality explanations and which faithfully relay the reasons behind the decisions of a model. Existing studies (DeYoung et al., 2020a; Adebayo et al., 2018, 2022; Ding and Koehn, 2021) point that explainability approaches can be unfaithful to the rationales used by a model and can cover potential flaws and biases in the model's reasoning. Such findings call for systematic diagnostics of explainability approaches to estimate their reliability and to motivate further progress in explainability approaches.

1.1.3.1 Explainability Diagnostics

Explainability approaches can be assessed by human judges that estimate the utility of explanations, e.g., for guessing the label predicted by the model (Lertvittayakumjorn and Toni, 2019; Narayanan et al., 2018). However, human studies often suffer from low inter-annotator agreement as the evaluation protocols can be subjective and assess how appealing explanations are to human judges instead of evaluating their qualities.

Another way of evaluating the utility of explainability approaches is by using automated measures for various explanation properties. One commonly assessed explainability property is faithfulness. It estimates whether an explainability approach faithfully reflects the rationales used in the decision-making process of a model. While some existing work (Alvarez-Melis and Jaakkola, 2018; Kindermans et al., 2019) estimates the lack of faithfulness based on a few counter-examples, Jacovi and Goldberg (2020) recommend the use of faithfulness evaluation measures, as in DeYoung et al. (2020a), computing rather a degree of explanation faithfulness. Another common evaluation measure is the extent of the explanation's agreement with human rationales, which indicates the plausibility and appeal of the rationales to human judges. DeYoung et al. (2020a); Ding and Koehn (2021) include measures of faithfulness, human agreement, and others in benchmarks for saliency-based explanations.

While some existing studies (Yin et al., 2022; Arras et al., 2019; Guan et al., 2019) evaluate explainability approaches with various measures for explainability properties, most studies are limited in scope, exploring only one or a few properties, datasets, and models. In *Paper 6* (§7) of this thesis, I construct a comprehensive list of diagnostic properties tied with automated measures thereof. The study provides a broad overview and a unified comparison of different groups of common explainability approaches across three text classification tasks and three model architectures. Stemming from the individual property results, the central finding of this work is that gradient-based methods have the best performance across all of the models and downstream text classification tasks considered in this work. Other explainability techniques, such as Shapley Value Sampling (Castro et al., 2009), LIME (Ribeiro et al., 2016a), and Occlusion (Zeiler and Fergus, 2014) take more time to compute, are considerably less faithful to the models, and are less consistent for similar model rationales and similar instances.

As saliency explanations provide a score for each input segment, there is a direct mapping between the explanation and the input. The latter enables explainability evaluation measures based on the mapping between the input and explanation. The same measures cannot be applied to NLEs as they contain words and rationales not explicitly present in the input. To this end, NLEs are usually evaluated using simulatability studies (Hase et al., 2020; Chan et al., 2022), where humans or models verify that the explanation indicates the label predicted by the model. Wiegreffe et al. (2021) also evaluate whether models generating NLEs pay attention to the same input tokens as the prediction model and whether model predictions and generated explanations are equally robust to noise introduced in the input.

1.1.3.2 Diagnostic-Guided Explainability

Besides evaluating the qualities of existing explainability approaches, diagnostic properties of explanations can also motivate the development of explainability techniques where those properties are improved. In *Paper 7* (§8) of this thesis, I present the first method that produces property-optimised explanations in an unsupervised way. As a result, the generated explanations have improved faithfulness to the underlying prediction method, they better indicate the confidence of the model's prediction and are more consistent across similar instances. A later study proposes an explainability approach that has improved sensitivity to adversarial perturbations of important tokens and is more consistent across similar instances (Yin et al., 2022). In addition, Thayaparan et al. (2022) propose novel differentiable combinatorial solvers that encode property constraints for explainable multi-hop inference.

1.1.4 Complex Reasoning in Natural Language Tasks

There has been substantial progress in natural language processing of downstream tasks where the input has a short textual form and requires a shallow-level semantic understanding of literal cues (Wang et al., 2018). Notably, we have witnessed the emergence of efficient natural language processing models that can be employed to automate a wide range of these tasks (Devlin et al., 2019; Liu et al., 2019). While such models can reach near-human performance on these tasks, shallow-level semantic understanding of literal cues is insufficient for many real-world natural language processing application tasks. Many real-world tasks, such as fact checking and question answering, require a human to possess a broad range of complex reasoning skills. Consequently, the current prevailing hypothesis in the field of natural language processing is that models need to possess similar reasoning skills to automate these real-world tasks. To achieve progress on these tasks and natural language processing in general, new benchmarks with tasks that constitute a more rigorous test of language understanding have been proposed (Wang et al., 2019).

Some examples of complex reasoning skills include reading comprehension (Rajpurkar et al., 2016), multi-hop composition (Yadav et al., 2019; Jiang et al., 2020), and logical reasoning (Liu et al., 2020a). Reading comprehension is the ability to deeply understand long-form textual input and locate the relevant text spans needed for correct inference. In some real-world scenarios, reading comprehension also involves the ability to detect when the provided text is missing information pertinent to drawing an inference. Building on reading comprehension skills, multi-hop composition incorporates the requirement for a model to find arguments scattered across multiple paragraphs or documents and connect them logically into a meaningful structure of arguments, e.g., a graph, that results in a correct prediction. Logical reasoning requires the model to deduce the logical relationship between statements in two textual inputs. Examples of logical reasoning are comparison, negation, categorical reasoning, disjunctive reasoning, and conjunctive reasoning.

Complex reasoning tasks require a model to obtain a combination of different complex reasoning skills to draw a correct inference. What follows next is a brief introduction to complex reasoning tasks central to this thesis.

1.1.4.1 Fact Checking

Fact checking is a time-consuming and elaborate task performed by human fact checkers. Automating the process is of pivotal importance for scaling the number of verified claims in accordance with the growing amount of misinformation and disinformation online. Most of the existing work on automating fact checking is concerned with predicting the veracity of a claim given evidence information (Ma et al., 2018; Mohtarami et al., 2018; Xu et al., 2018; Augenstein et al., 2019).

The *reasoning skills* required for automatic fact checking of claims depend on the nature of the employed dataset. Artificially constructed datasets (Thorne et al., 2018; Schuster et al., 2021), where claims have been written based on Wikipedia evidence, can involve handling negations and simple lexical and semantic matching between the evidence and the claims. Some of them are designed to test for specific skills such as multi-hop reasoning (Jiang et al., 2020) and tabular reasoning over structured evidence from tables in Wikipedia (Aly et al., 2021). Fact checking datasets containing real claims and evidence can require various complex reasoning skills, including multi-hop, logical and mathematical reasoning, but are limited in size (Alhindi et al., 2018; Kotonya and Toni, 2020a).

Existing work has explored the *accountability* of fact checking models and pointed to the following model deficiencies. Schuster et al. (2019) were the first to reveal that fact checking models often make predictions based solely on the claim without consulting the provided evidence. Schuster et al. (2021) show that fact checking models exhibit a bias for significantly higher word overlap in supporting evidence-claim pairs over refuting pairs. Finally, Atanasova et al. (2022) (*Paper 1* §2) point that fact checking models are prone to make predictions based on insufficient information. Thorne et al. (2019a) are the first to propose hand-crafted adversarial attacks for fact checking systems. In the subsequent FEVER 2.0 task (Thorne et al., 2019b), participants designed adversarial attacks for existing fact checking systems testing for multi-hop reasoning (Niewinski et al., 2019; Hidey et al., 2020) or generated various attacks manually (Kim and Allan, 2019). Finally, Atanasova et al. (2020c) (*Paper 2* §3) propose a method to generate highly potent and semantically coherent adversarial attacks in an automated way.

Saliency explanations, abstractive, and extractive NLEs have been studied to enhance the *transparency* of fact checking systems. In Wikipedia-based datasets, sentences from Wikipedia documents are extracted as explanations, and the retrieval performance of systems is measured jointly with the verification task – a label prediction is considered correct only when the correct evidence is found (Thorne et al., 2018). In real-world datasets, where the claims are not artificially produced but occur naturally, summaries of long ruling comments justifying claims are used as explanations, and generated explanations are evaluated with ROUGE scores (Atanasova et al., 2020c; Kotonya and Toni, 2020a). The implementation of the methods producing NLEs varies widely and can be grouped into models optimised separately or jointly with the task at hand (Malon, 2018; Atanasova et al., 2020c) (*Papers 3* §4, *4* §5, *5* §6). Existing work also proposes differentiable theorem proving approaches, which are self-interpretable models providing logical relations between the evidence and the claim and leading to a prediction (Krishna et al., 2021).

1.1.4.2 Question Answering

Similar to fact checking, existing work has indicated that automatic question answering models have to learn a variety of complex reasoning skills, equivalent to human reasoning skills, in order to perform well on the task (Rogers et al., 2021; Choudhury et al., 2021). Challenge datasets are developed to audit question answering systems for complex reasoning skills such as multi-hop reasoning (Yadav et al., 2019), unanswerable questions (Rajpurkar et al., 2016), and logical reasoning (Liu et al., 2020a). Explanations for question answering systems are produced by extracting supporting sentences from the provided document (Yadav et al., 2020; Thayaparan et al., 2020) (*Paper 7* §8). Another way of producing explanations for question answering systems is by generating NLEs (Rajani et al., 2019), e.g., for common-sense multiple-choice question answering.

1.1.4.3 Natural Language Inference

Natural language inference is the task of recognising textual entailment (Dagan et al., 2013) between two pieces of text, namely the premise and the hypothesis. Models have to predict the relation between the two parts, which could be entailment, contradiction, or neutral. Multiple challenge datasets have been developed to audit the reasoning capabilities of natural language inference models, including linguistic (Saha et al., 2020) and logical reasoning (Tian et al., 2021). Other studies produce challenge datasets to reveal flaws in the reasoning of automatic natural language inference models. Gururangan et al. (2018) point that models can attain high performance based only on the premise without consulting the hypothesis. Sanchez et al. (2018) find that natural language inference models are insensitive to small but semantically significant changes, and that their predictions can be manipulated with simple statistical correlations between words and training labels present in the training split. Explainability approaches for natural language inference models include post-hoc saliency explanations as well as abstractive NLEs (Camburu et al., 2018), which usually are single sentences explaining the rationales of a model.

1.1.5 Modelling Complex Reasoning Tasks

Lately, language models (LMs) employing the Transformer architecture (Vaswani et al., 2017; Devlin et al., 2019) have become the core building blocks of architectures utilised to effectively automate many machine learning problems, including complex reasoning tasks. Such models have been the subject of rigorous studies inspecting their capabilities to learn complex reasoning skills. Talmor et al. (2020) find that different Transformer models exhibit qualitatively different reasoning abilities, such as that only RoBERTa-L (Liu et al., 2019), compared to BERT (Devlin et al., 2019) and other RoBERTa model sizes, performs well at number comparison. Kassner and Schütze (2020); Nie et al. (2020) find that LMs cannot detect the presence of negation in the input text. Talmor et al. (2020) discover that LMs are unable to learn multi-hop reasoning and even struggle to learn it with some supervision.

Several architectural improvements have been proposed to enhance the reasoning abilities of LMs. Graph attention networks (Liu et al., 2020c; Zhou et al., 2019) and eXtra-hop attention (Zhao et al., 2020) have been employed to improve the multi-hop reasoning abilities of LMs. Knowledge graphs have been incorporated into LMs to improve common-sense reasoning skills (Ilievski et al., 2021). Furthermore, contrastive learning techniques have been explored to improve a model's performance, especially given contrastive challenge datasets for particular skills (Schuster et al., 2021; Atanasova et al., 2022). The publications in this thesis consider the Transformer architecture and its extensions to handle multi-hop reasoning (*Paper 5*) as well as contrastive learning to improve a model's performance for instances with insufficient information (*Paper 1*).

1.2 Scientific Contributions

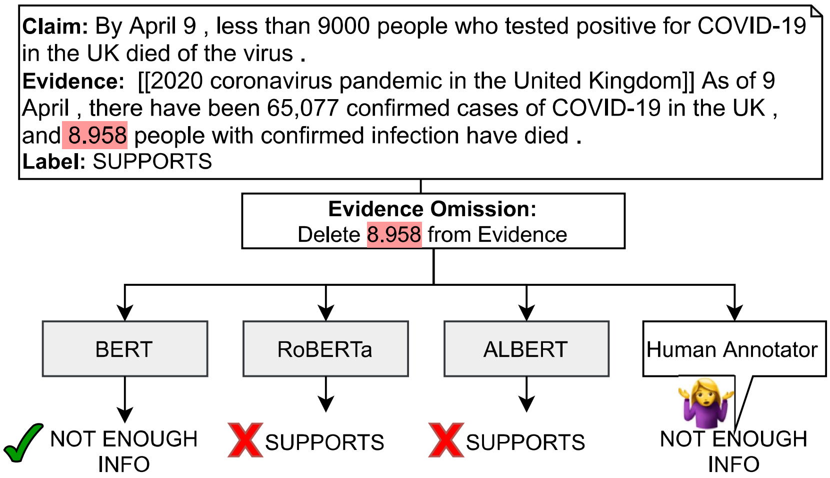

Claim: By April 9 , less than 9000 people who tested positive for COVID-19 in the UK died of the virus .
Evidence: [[2020 coronavirus pandemic in the United Kingdom]] As of 9 April , there have been 65,077 confirmed cases of COVID-19 in the UK , and 8.958 people with confirmed infection have died .
Label: SUPPORTS
Evidence Omission:
Delete 8.958 from Evidence
BERT
RoBERTa
ALBERT
Human Annotator
✓ NOT ENOUGH INFO
✗ SUPPORTS
✗ SUPPORTS
NOT ENOUGH INFO

Figure 1.1: An example from the VitaminC test set, where the number modifier has been omitted from the evidence. This results in insufficient evidence for predicting its support for the claim as judged by human annotators. Two of the models still find the remaining evidence to be sufficient.

1.2.1 Accountability for Complex Reasoning Tasks over Text

1.2.1.1 Paper 1: Fact Checking with Insufficient Evidence

Automating the fact checking process relies on information obtained from external sources (Thorne et al., 2018; Leippold and Diggelmann, 2020; Augenstein, 2021) (see Section 1.1.4.1). However, the necessary information is not always available, either due to incomplete knowledge sources, or because the claim has newly emerged and the relevant facts are not documented yet. In this work, I posit that it is crucial for fact checking models to make veracity predictions only when there is sufficient evidence and otherwise indicate when it is not enough.

To this end, this work introduces the **novel task of Evidence Sufficiency Prediction illustrated in Figure 1.1 , which is defined as the task of identifying what information is sufficient for making a veracity prediction by fact checking models.** I study the new task by, first, conducting a thorough empirical analysis of what models consider to be sufficient evidence for fact checking. For the **empirical analysis**, I propose **a new fluency-preserving method that occludes portions of the evidence**, automatically removing constituents or entire sentences, to create incomplete evidence. Secondly, I collect human annotations for sufficient evidence for fact checking, which results in a **novel challenge dataset, *SufficientFacts***, for fact checking with omitted evidence. I observe that it is the hardest for fact checking models to detect when the evidence is missing information for the prediction that was removed from adverbial modifiers, followed by subordinate clauses. By contrast, it is easiest to detect missing

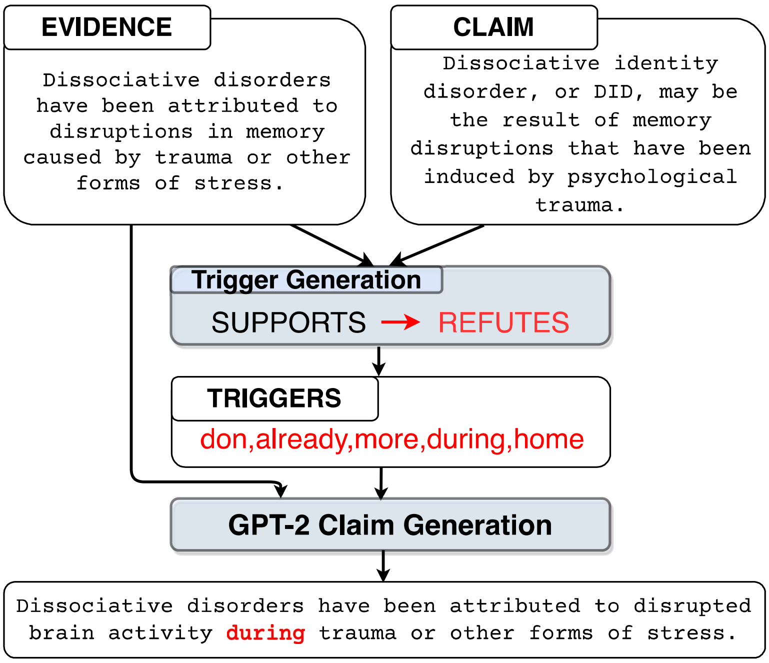

EVIDENCE
Dissociative disorders have been attributed to disruptions in memory caused by trauma or other forms of stress.
CLAIM
Dissociative identity disorder, or DID, may be the result of memory disruptions that have been induced by psychological trauma.
Trigger Generation
SUPPORTS → REFUTES
TRIGGERS
don,already,more,during,home
GPT-2 Claim Generation
Dissociative disorders have been attributed to disrupted brain activity during trauma or other forms of stress.

Figure 1.2: High level overview of the method. First, universal adversarial triggers are discovered for flipping a source to a target label (e.g. SUPPORTS → REFUTES). These triggers are then used to condition the GPT-2 language model to generate novel claims with the original label, including at least one of the found triggers.

information when it is a date modifier, followed by number modifiers. Finally, I employ the information occlusion method introduced for the empirical analysis to **improve the performance of models on the new task of Evidence Sufficiency Prediction**. I show that considering it a component task of fact checking significantly improves fact checking performance. The performance for Evidence Sufficiency Prediction is improved by up to 17.8 F_1 score, which in turn improves fact checking performance by up to 2.6 F_1 score.

1.2.1.2 Paper 2: Fact Checking with Insufficient Evidence

Adversarial attacks reveal vulnerabilities and flaws of trained models (Goodfellow et al., 2014; Szegedy et al., 2013). One attack type that has a high success rate in fooling a provided machine learning model is the universal adversarial triggers approach (Wallace et al., 2019a). It produces individual n-grams, termed triggers, that, when appended to instances of a class under attack, can trick a model into predicting a target class, different from the instances' correct labels. However, for inference tasks such as fact checking, these triggers often invert the meaning of instances they are inserted into, thus also changing their gold labels. In addition, such attacks produce nonsensical inputs, as they simply concatenate triggers to existing samples. This paper proposes to address these two deficiencies of universal adversarial attacks, thus allowing for automatically generated adversarial attacks against fact checking systems that are both semantically valid and have correct gold labels.

The core contribution of the paper is a **method for automatically generating potent adversarial examples** that **preserve the meaning** of the source text and **improve the semantic validity** of universal adversarial triggers. This is accomplished via: 1) a **novel extension to the HotFlip attack** (Ebrahimi et al., 2018), which jointly minimizes the target class loss of a fact checking model and the entailment class loss of a natural language inference model; 2) a **conditional language model** trained using GPT-2 (Radford et al., 2019), which takes trigger tokens and a piece of evidence, and generates a semantically coherent new claim containing at least one trigger. Figure 1.2 shows an overview of the method. The resulting triggers maintain potency against a fact checking model while preserving the original claim label. Moreover, the conditional language model produces semantically coherent adversarial examples containing triggers, which lead to a decrease of 23.1 F_1 score points in the performance of the fact checking model when compared to its performance on the original claims. The resulting adversarial attacks unveil the vulnerability of fact checking models to particular trigger words present in the input, which require the development of appropriate defenses for ensuring robust fact checking performance.

1.2.2 Explainability for Complex Reasoning Tasks over Text

1.2.2.1 Paper 3: Generating Fact Checking Explanations

Most existing work on automated fact checking is concerned with predicting the veracity of claims based on metadata, social network spread, language used in claims (Ma et al., 2018; Mohtarami et al., 2018; Xu et al., 2018), and, more recently, evidence supporting or denying claims (Thorne et al., 2018; Stammbach and Neumann, 2019). A crucial piece of the puzzle that is still missing is to understand how to automate the most elaborate part of the process – generating justifications for verdicts on claims.

In this publication, I present the **first study on generating natural language veracity explanations**, showing that they can successfully describe the reasons behind a veracity prediction as illustrated in Table 1.1. This work frames fact checking explanations as extractive summarisation to address the challenges of the task stemming from the complex reasoning required for claim verification and the limited training data. I find that the veracity prediction model can utilise information from the detailed fact checking reports of professional journalists, resulting in a performance increase. The performance can be further improved by training veracity prediction and veracity explanation jointly. Notably, optimising the joint objective of veracity prediction and

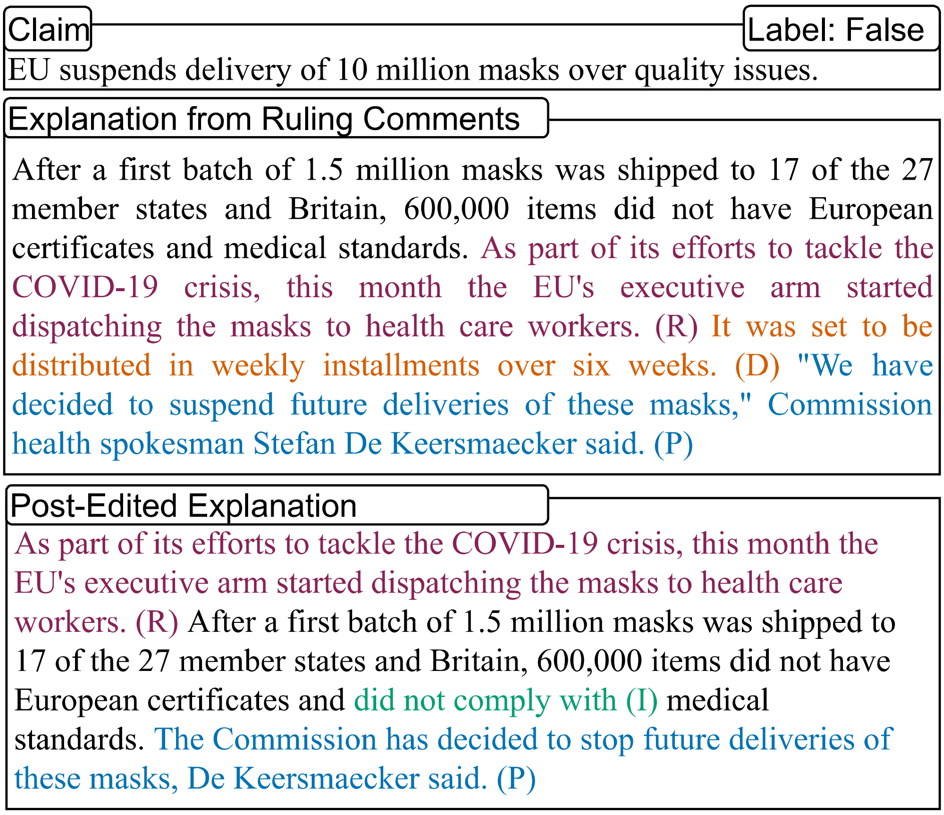
Claim: The last major oil spill from a drilling accident in America happened over 40 years ago in 1969.
Ruling Comments: (...) The last major oil spill from a drilling accident in America happened over 40 years ago in 1969. (...) The largest in volume was the Santa Barbara spill of 1969 referenced by Murdock and Johnson, in which an estimated 100,000 barrels of oil spilled into the Pacific Ocean, according to the API. The Santa Barbara spill was so big it ranked seventh among the 10 largest oil spills caused by marine well blowouts in the world, the report states. Two other U.S. spills, both in 1970, rank eighth and 10th. Fourteen marine blowouts have taken place in the U.S. between 1969 and 2007. Six of them took place after 1990 and spilled a total of nearly 13,700 barrels. (...) We interviewed three scientists who said that the impact of a spill has little to do with its volume. Scientists have proven that spills far smaller than Santa Barbara's have been devastating.
Justification: While the nation's largest oil well blowout did take place in 1969, it's not factually correct to call it the "last major oil spill". First of all, two of the largest blowouts in the world took place in the U. S. the following year. More importantly, experts agree that spills far smaller in volume to the 1969 disaster have been devastating. From a scientific perspective, Johnson's decision to single out the 1969 blowout as the last "major" one makes no sense.
Ruling: Half-True

Table 1.1: Example instance from the LIAR-PLUS dataset, with oracle sentences for generating the justification highlighted.

veracity explanation produces explanations that achieve better coverage and overall quality and serve better at explaining the correct veracity label than explanations learned solely to mimic human justifications. Overall, this work establishes important fundamentals in the area of transparency for models where explanations require complex reasoning and consist of multiple arguments spanning over several sentences and where the training resources are limited.

1.2.2.2 Paper 4: Generating Fluent Fact Checking Explanations with Unsupervised Post-Editing

Fact-checking systems have become important tools to verify fake and misguiding news. These systems become more trustworthy when human-readable explanations accompany the veracity labels. Prior work (Atanasova et al., 2020c) (Paper 3 §4) has proposed to use automatic summarisation to select a subset of sentences from the long ruling comments (RCs) of professional journalists and used them as short layman explanations for fact checking veracity predictions. However, with a purely extractive



Claim

Label: False

EU suspends delivery of 10 million masks over quality issues.

Explanation from Ruling Comments

After a first batch of 1.5 million masks was shipped to 17 of the 27 member states and Britain, 600,000 items did not have European certificates and medical standards. As part of its efforts to tackle the COVID-19 crisis, this month the EU's executive arm started dispatching the masks to health care workers. (R) It was set to be distributed in weekly installments over six weeks. (D) "We have decided to suspend future deliveries of these masks," Commission health spokesman Stefan De Keersmaecker said. (P)

Post-Edited Explanation

As part of its efforts to tackle the COVID-19 crisis, this month the EU's executive arm started dispatching the masks to health care workers. (R) After a first batch of 1.5 million masks was shipped to 17 of the 27 member states and Britain, 600,000 items did not have European certificates and did not comply with (I) medical standards. The Commission has decided to stop future deliveries of these masks, De Keersmaecker said. (P)

Figure 1.3: Example of a post-edited explanation from PubHealth that was initially extracted from RCs. It illustrates four post-editing steps: reordering (R), insertion (I), deletion (D), and paraphrasing (P).

approach, the sentences are cherry-picked from different parts of the corresponding RCs, and as a result, explanations are often disjoint and non-fluent.

This work presents an **iterative edit-based algorithm that only uses phrase-level edits to perform unsupervised post-editing of disconnected extractive explanations** as illustrated in Figure 1.3. To the best of my knowledge, this work is the first to explore an iterative unsupervised edit-based algorithm using only phrase-level edits. The proposed algorithm also leads to the first computationally feasible solutions for unsupervised post-editing of long text inputs. A scoring function with components including fluency and semantic preservation is used to regulate the editing algorithm. Notably, combining the iterative post-editing algorithm with grammatical correction and paraphrasing-based post-processing leads to fluent and easy-to-read explanations. The paper presents extensive experiments on the LIAR-PLUS (Wang, 2017) and PubHealth (Kotonya and Toni, 2020a) fact checking datasets. The automated evaluation confirms the success of the proposed method for preserving the semantics important to perform verification of the claim and enhancing the readability of the

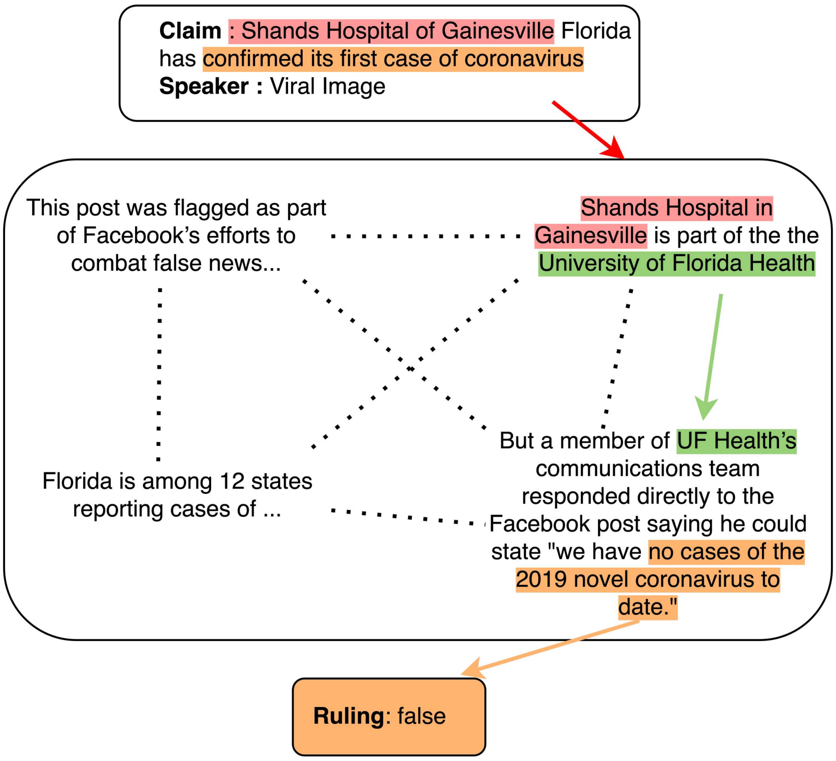

Claim : Shands Hospital of Gainesville Florida has confirmed its first case of coronavirus
Speaker : Viral Image
This post was flagged as part of Facebook's efforts to combat false news...
Shands Hospital in Gainesville is part of the the University of Florida Health
Florida is among 12 states reporting cases of ...
But a member of UF Health's communications team responded directly to the Facebook post saying he could state "we have no cases of the 2019 novel coronavirus to date."
Ruling: false

Figure 1.4: An illustration of multiple hops over an instance from `PolitiHop`. Each instance consists of a claim, a speaker, a veracity label, and a PolitiFact article the annotated evidence sentences. The highlighted sentences represent the evidence sentences a model needs to connect to arrive at the correct veracity prediction.

generated explanations. Finally, a manual evaluation confirms that the proposed approach improves the fluency and conciseness of the generated explanations.

1.2.2.3 Paper 5: Multi-Hop Fact Checking of Political Claims

As noted in Section 1.1.4.1, one of the important reasoning skills required for fact checking is multi-hop reasoning, where a set of connected evidence pieces leads to the final verdict of a claim, as illustrated in Figure 1.4. However, existing datasets do not provide annotations for gold evidence pages, except for FEVER (Thorne et al., 2018), where only 17% of the claims require multi-hop reasoning and the claims are constructed artificially.

This publication presents a study of more complex **claim verification with naturally occurring claims where rationales consist of multiple hops over interconnected evidence chunks**. To the best of my knowledge, this is the first work on multi-hop fact checking of political claims. To this end, this study constructs a small annotated dataset, PolitiHop, of evidence sentences for claim verification for the task. PolitiHop is employed to analyze to what degree existing multi-hop reasoning methods are suitable for the task. Furthermore, PolitiHop is used to investigate whether reasoning skills

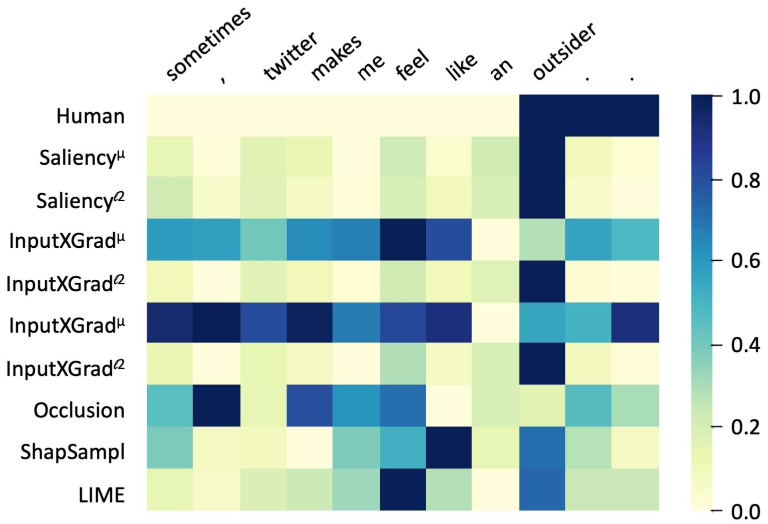


Figure 1.5: Example of the saliency scores for the words (columns) of an instance from the Twitter Sentiment Extraction dataset. They are produced by the explainability techniques (rows) given a `Transformer` model. The first row is the human annotation of the salient words. The scores are normalized in the range $[0, 1]$.

learned with a multi-hop model on similar datasets can be transferred to PolitiHop. The main finding of the study is that the task of multi-hop fact checking of real-world claims is complex and that the best performance is achieved with an architecture that specifically models multi-hop reasoning over evidence pieces in combination with in-domain transfer learning.

1.2.3 Diagnostic Explainability Methods

1.2.3.1 Paper 6: A Diagnostic Study of Explainability Techniques for Text Classification

Recent developments in machine learning have introduced models that approach human performance at the cost of increased architectural complexity (Strubell et al., 2019). Efforts to make the rationales behind the models' predictions transparent have inspired an abundance of new explainability techniques. Provided with an already trained model, they compute saliency scores for the words of an input instance as illustrated in Figure 1.5 (see Section 1.1.2). However, there exists no definitive guide for: (i) how to choose such a technique given a particular application task and model architecture; and (ii) the benefits and drawbacks of using each such technique. In this paper, I develop a comprehensive list of diagnostic properties for evaluating existing explainability techniques.

This work presents a **comprehensive list of diagnostic properties for explainability and automatic measurement of them**, allowing for their effective assessment

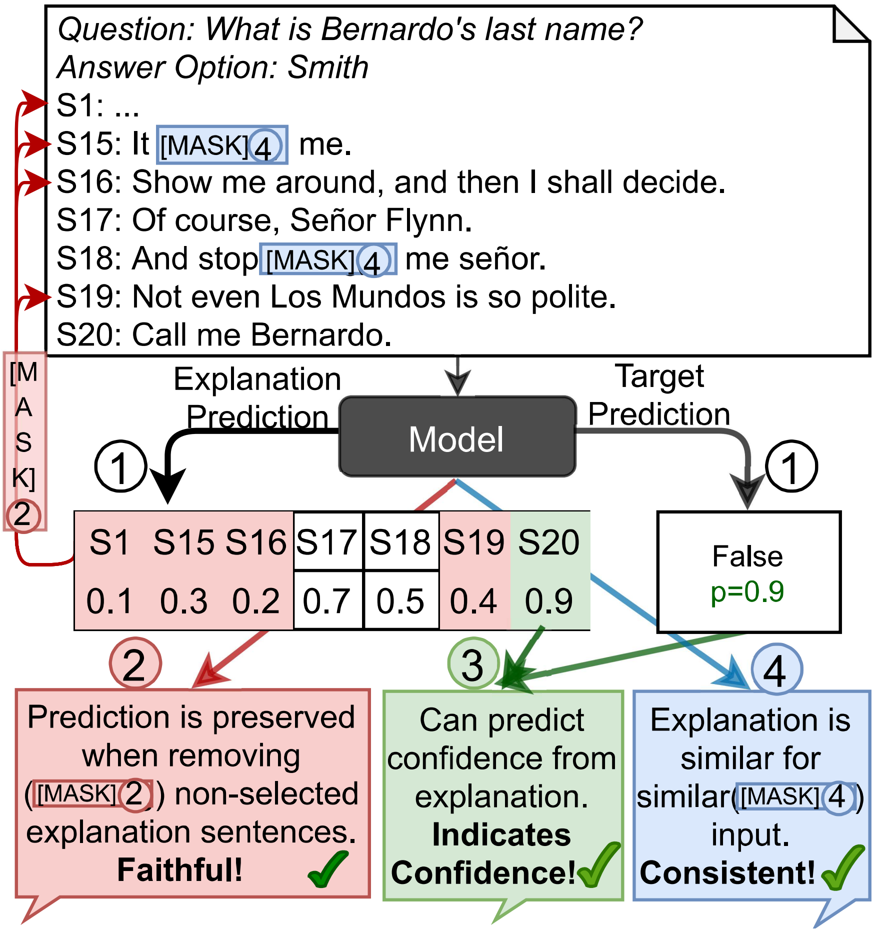

Question: What is Bernardo's last name?
Answer Option: Smith
S1: ...
S15: It [MASK] 4 me.
S16: Show me around, and then I shall decide.
S17: Of course, Señor Flynn.
S18: And stop [MASK] 4 me señor.
S19: Not even Los Mundos is so polite.
S20: Call me Bernardo.
[MASK] 2
Explanation Prediction
Model
Target Prediction
1
1
S1 S15 S16 S17 S18 S19 S20
0.1 0.3 0.2 0.7 0.5 0.4 0.9
False
p=0.9
2
3
4
Prediction is preserved when removing ([MASK] 2) non-selected explanation sentences. Faithful!
Can predict confidence from explanation. Indicates Confidence!
Explanation is similar for similar([MASK] 4) input. Consistent!

Figure 1.6: Example instance from MultiRC with predicted target and explanation (Step 1), where sentences with confidence ≥ 0.5 are selected as explanations (S17, S18, S20). Steps 2-4 illustrate the use of Faithfulness, Data Consistency, and Confidence Indication diagnostic properties as additional learning signals. '[MASK](2)' is used in Step 2 for sentences (in red) that are not explanations, and '[MASK](4)'–for random words in Step 4.

in practice. The proposed list of diagnostic properties is used to study and compare the characteristics of different groups of explainability techniques in three different application tasks and three different model architectures. Furthermore, the list of diagnostic properties is employed to study the attributions of the explainability techniques and human annotations of salient regions to compare and contrast the rationales of humans and machine learning models. Notably, the main finding of this diagnostic study of explainability techniques is that the investigated gradient-based explanation generation methods perform best across tasks and model architectures. This work also presents further detailed insights into the properties of the reviewed explainability techniques.

1.2.3.2 Paper 7: Diagnostics-Guided Explanation Generation

Extractive natural language explanation techniques shed light on a machine learning model's rationales by producing free text explanations, which are usually whole sentences extracted from the input (§1.1.2.2). Such techniques are typically constructed as models trained in a supervised way given human explanations. When such annotations are not available, explanations are often selected as those portions of the input that maximise a downstream task's performance, which corresponds to optimising an explanation's faithfulness to a given model. Faithfulness is one of several so-called diagnostic properties, which prior work has identified as useful for gauging the quality of an explanation without requiring annotations (DeYoung et al., 2020a). Other diagnostic properties are Data Consistency, which measures how similar explanations are for similar input instances, and Confidence Indication, which shows whether the explanation reflects the confidence of the model (Atanasova et al., 2021) (Paper 6 §7).

The main contribution of this paper is a **novel method to learn the diagnostic properties – Faithfulness, Data Consistency, and Confidence Indication, in an unsupervised way**, directly optimising for them to improve the quality of generated explanations as illustrated in Figure 1.6. I implement a joint task prediction and explanation generation model, which selects rationales at sentence level. Each property can then be included as an additional training objective in the joint model. With experiments on three complex reasoning tasks, I find that apart from improving the properties I optimised for, diagnostic-guided training also leads to explanations with higher agreement with human rationales and improved downstream task performance. Moreover, I find that jointly optimising for diagnostic properties leads to a reduced claim/question-only bias (Schuster et al., 2019) for the target prediction, which means that the model relies more extensively on the provided evidence. Importantly, I also find that optimising for diagnostic properties of explanations without supervision for explanation generation does not lead to good human agreement. This indicates the need for human rationales to train models that make the right predictions for the right reasons.

1.3 Summary of Contributions and Future Work

The publications in this thesis collectively contribute to advancing the state of the art of accountable and transparent machine learning for complex reasoning tasks over text. In particular, they facilitate the analysis of the reasons behind the outputs of ML

	Accountability			Explainability			Explainability Diagnostics		
	M	D	A	M	D	A	M	D	A
1. Atanasova et al. (2022)	✓	✓	✓						
2. Atanasova et al. (2020c)	✓	✓	✓						
3. Atanasova et al. (2020b)				✓					
4. Jolly et al. (2021)				✓					
5. Ostrowski et al. (2021)		✓	✓		✓	✓			
6. Atanasova et al. (2020a)							✓		✓
7. Atanasova et al. (2021)				✓			✓		

Table 1.2: Summary of contributions made by the publications in this thesis by topic – Accountability Methods, Explainability and Explainability Diagnostics, and type of contribution – Methodological (M), Dataset (D), Diagnostic Analysis (A).

models and assist in detecting and correcting for potential harms. Table 1.2 maps the methodological, dataset, and analysis contributions of each paper along each of the accountability and transparency axes.

Many of the proposed methods for auditing and explaining machine learning models in this thesis are empirically validated on the task of fact checking as it provides a rich test bed for testing complex reasoning over text. Fact checking is also considered in the publications in this thesis as it is particularly critical when developing models for this task that they are both accountable and transparent. I further test on other complex reasoning tasks, namely natural language inference and question answering, where appropriate datasets are available. Due to the fact-checking task's complexity and the methods proposed in this work being generally applicable, they could also be easily validated on other tasks requiring complex reasoning skills, given suitable benchmark datasets.

1.3.1 Accountability for Complex Reasoning Tasks over Text

The contributions made in the area of *accountability* of machine learning models include challenge datasets for prediction with insufficient information (Paper 1 §2) and multi-hop reasoning (Paper 5 §6) as well as a dataset showing a model's vulnerability to adversarial manipulations (Paper 2 §3). The methodological contributions presented in my thesis enable the automated generation of these challenge and adversarial datasets, which extends the applicability of the proposed accountability audits for other complex reasoning tasks and machine learning models. The resulting resources reveal important insights about the models' capabilities and advance our understanding of the models' decision-making processes. Furthermore, they reveal model vulnerabilities that necessitate the development of appropriate complex reasoning models and defenses against adversarial attacks. To this end, this thesis also makes methodological contributions that improve the reasoning capabilities of machine learning models regarding the uncovered vulnerabilities, thus leading to enhancements in their accountability.

While challenge and adversarial datasets for complex reasoning are developed mainly for natural language inference, I present studies improving the accountability of fact checking systems, which have longer textual inputs and require more complex reasoning skills with compositions thereof. Moreover, the accountability of fact checking systems in deployment is imperative and requires extensive research, even more so for critical domains such as fact checking of medical claims.

The current research landscape of challenge datasets and adversarial attacks for complex reasoning tasks is discordant – separate studies investigate a limited number of complex reasoning skills and reasoning flaws over one or a few models and tasks. Hence, some prospects for future work include efforts to unify the studies and findings on complex reasoning skills and flaws in models' rationales across different tasks and models. This could potentially result in benchmarks designed around the notion of skills rather than downstream tasks, which could facilitate overall assessments of the accountability of machine learning models. Furthermore, future interdisciplinary synergies involving linguistics, cognitive science, and machine learning could lead to the design of comprehensive lists of complex reasoning skills.

1.3.2 Explainability for Complex Reasoning Tasks over Text

In regards to *model transparency*, this thesis pushes the state-of-the-art for generated natural language explanations (Papers 3 §4, 4 §5, and 5 §6, and 7 §8) for fact checking systems. Generating NLEs for the veracity predictions of real-world claims is a particularly challenging task as there are limited training resources, and it requires multiple connected arguments to be presented in a readable and accessible way. With this thesis, I lay the foundations for automatically generating such explanations. The produced explanations improve our understanding of fact checking models' decision-making processes. They can instill trust in the models' predictions, serve as a further means for auditing the accountability of machine learning models, and enable end users to expand their knowledge by leveraging information from the rationales of the models.

There is still a lack of sufficiently large datasets for generating complex reasoning explanations for real-world fact checking and, in general, for explanations consisting of multiple arguments spanning over several sentences. This limits the possible achieved quality of generated explanations and could be addressed in future work. Moreover, there is a need for datasets that could allow for automating real-world fact checking explainability fully – from collecting evidence documents to producing veracity labels and explanations.

There are many prospects for enhancing the availability of datasets for natural language explanation generation in general. The explanations contained in existing natural language inference datasets are often based on templates making them rather structure-based explanations (Wiegreffe and Marasovic, 2021) and the quality of some common-sense NLE datasets is questioned in related work (Narang et al., 2020; Wiegreffe and Marasovic, 2021). Moreover, there are currently no existing datasets with NLEs for tasks that require different complex reasoning skills. Such datasets could be developed in future work to generate explanations and study models' rationales for the different types of complex reasoning skills required for a task.

1.3.3 Diagnostic Explainability Methods

Explainability techniques aim to reveal the rationales of machine learning models. End users make decisions to trust and rely on models' predictions based on the explanations produced to reveal the rationales employed by the models for their predictions. Hence, explainability techniques have to be robust and faithful to the underlying model as well. This thesis moves forward our collective knowledge of the field of *explanation diagnostics* with a diagnostic study of post-hoc saliency-based explainability techniques (Paper 6 §7), which are further directly optimised for in generated explanations, thus improving explanation quality (Paper 7 §8). The insights gained from the analysis performed in my thesis reveal which explainability techniques perform better than others, as well as which tasks and models necessitate the development of more robust and appropriate explainability techniques. Finally, improving the quality of the produced explanations additionally enhances the understanding and trust in the model's rationales. Measuring and improving explanations' quality instills trust in the employed explainability approach as well.

Currently, there is limited work on measuring and ensuring the quality of NLEs (Wiegreffe et al., 2021; Hase et al., 2020). NLEs are generated by supervised systems, optimised to resemble human-annotated explanations, which does not guarantee that they convey the rationales used by a model. This calls for future studies examining the faithfulness and other properties of NLEs and improving these properties in generated NLEs.

Part II

Accountability for Complex Reasoning Tasks over Text

Fact Checking with Insufficient Evidence

2

2.1 Introduction

Computational fact checking approaches typically use deep learning models to predict the veracity of a claim given background knowledge (Thorne et al., 2018; Leippold and Diggelmann, 2020; Augenstein, 2021). However, the necessary evidence is not always available, either due to incomplete knowledge sources, or because the claim has newly emerged and the relevant facts are not documented yet. In such cases, FC models should indicate that the information available is insufficient to predict the label, as opposed to making a prediction informed by spurious correlations.

Prior work shows that FC models can sometimes predict the correct veracity based on just the claim, ignoring the evidence, and that they can overly rely on features such as the word overlap between the evidence and the claim (Schuster et al., 2019, 2021), leading to biased predictions. However, there are no previous studies on what evidence a FC model considers to be enough for predicting a veracity label. To this end, this work introduces the **novel task of Evidence Sufficiency Prediction illustrated in Fig. 2.1 , which we define as the task of identifying what information is sufficient for making a veracity prediction.** This task is related to FC and can operate on instances and models from FC datasets, but is focused on evaluating the capability of models to detect missing important information in the provided evidence for a claim. The latter is usually not evaluated explicitly in current FC benchmarks, where joint scores disregard a FC model's prediction when insufficient evidence is retrieved.

We study the new task by, first, conducting a thorough empirical analysis of what models consider to be sufficient evidence for FC. Secondly, we collect human annotations for the latter, which results in a novel diagnostic dataset, *SufficientFacts*, for FC with omitted evidence. Finally, we employ the method introduced for the empirical analysis to improve the performance of models on the new task of Evidence Sufficiency Prediction, and show that considering it a component task of FC significantly improves FC performance.

For the **empirical analysis**, we propose a new fluency-preserving method that occludes portions of evidence, automatically removing constituents or entire sentences, to create incomplete evidence. We provide those as input to an ensemble of Transformer-based FC models to obtain instances on which FC models agree vs. disagree to have (in)sufficient information. We perform extensive experiments with three

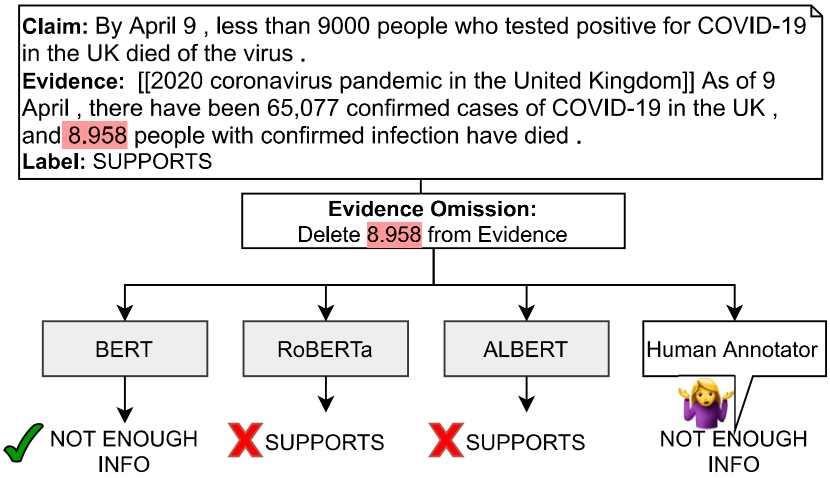

Claim: By April 9 , less than 9000 people who tested positive for COVID-19 in the UK died of the virus .
Evidence: [[2020 coronavirus pandemic in the United Kingdom]] As of 9 April , there have been 65,077 confirmed cases of COVID-19 in the UK , and 8.958 people with confirmed infection have died .
Label: SUPPORTS
Evidence Omission:
Delete 8.958 from Evidence
BERT
RoBERTa
ALBERT
Human Annotator
✓ NOT ENOUGH INFO
✗ SUPPORTS
✗ SUPPORTS
NOT ENOUGH INFO

Figure 2.1: An example from the VitaminC test set, where the number modifier has been omitted from the evidence. This results in there not being enough evidence for predicting its support for the claim as judged by human annotators, while two of the models still find the remaining evidence to be sufficient.

models – BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and three textual FC datasets with different types of claims – FEVER (Thorne et al., 2018), HoVer (Jiang et al., 2020), VitaminC (Schuster et al., 2021).

To compare model behavior with human rationales for Evidence Sufficiency Prediction, we ask annotators to indicate if the occluded evidence texts still provide enough information for a fact-check. This results in a **novel diagnostic test dataset, *SufficientFacts***, which contains information about the type of the omitted information, allowing for in-depth analyses of model behavior.

Finally, to improve model performance for detecting omitted important evidence and, in turn, FC, we propose to combine the proposed evidence omission method with tri-training (Zhou and Li, 2005), which utilises the agreement of three different machine learning models to label unlabeled training instances (§2.5). This results in a **novel counterfactual data augmentation schema for learning of (in)sufficient information**. We find that the proposed approach is highly effective in improving model performance by up to 17.8 F_1 score on the newly introduced *SufficientFacts*. This also leads to improvements of up to 2.6 F_1 score on the standard FC test sets for the corresponding datasets.

2.2 Related Work

Here, we study when models trained on existing FC datasets find evidence with omitted important information to still be sufficient for veracity prediction. Such cases might be considered vulnerabilities of the models and can be due to models' faulty reasoning, learned biases, etc. Hence, our work is mainly related to studies exploring potential biases learned by FC models and the vulnerabilities of FC models to adversarial attacks. We further propose a method for evidence omission, which creates counterfactual instances, which is related to studies on input-level instance re-writing. We also use the proposed evidence omission method to collect counterfactually augmented data (CAD) and compare that to using the collected data in a contrastive learning (CL) loss to improve performance on Evidence Sufficiency Prediction and FC more generally. We thus discuss the relationship between our work and prior studies on CAD and CL. Finally, we compare our work based on deep learning models to FC performed against knowledge bases (KBs), where fact triples can also be missing.

Fact Checking Diagnostics. Previous work has exposed various biases of FC models. While FEVER (Thorne et al., 2018) is one of the largest datasets for FC, Schuster et al. (2019) points out that models trained on it can verify a claim solely based on the text of the claim, without considering the evidence. To this end, Schuster et al. (2019) introduce a new diagnostic dataset, FeverSymmetric, of contrastively re-written claims and evidence. They show that the models fail to detect the contrastive changes in the text, leading to a drop of up to 57.46 F_1-score, compared to 85.85 F_1-score on the original FEVER development set. Furthermore, the claims in FEVER were manually written based on Wikipedia article sentences, and thus have a large token overlap between the evidence and the claim, especially for supporting evidence. Hence, Schuster et al. (2021) construct a new FC dataset, VitaminC, where they instruct the annotators to avoid using the same words as in the evidence. Ostrowski et al. (2021) further create PolitiHop – a dataset for claim verification of naturally occurring claims with evidence comprised of multiple hops over interconnected evidence chunks. They study how multi-hop vs. single inference architectures reason over the evidence sets in PolitiHop. In addition, several works (Thorne et al., 2019a; Niewinski et al., 2019; Hidey et al., 2020) explored the vulnerability of FC models to adversarial attacks, e.g., by discovering universal trigger words that fool a model into wrongly changing its prediction (Atanasova et al., 2020c). In contrast, we are interested in how much evidence is enough for veracity prediction, studying this with three different FC models trained on three different datasets by omitting information at the constituent and sentence levels and comparing it to human judgments.

Instance Re-Writing. The above studies mainly perform re-writing or insertion operations for FC evidence. Here, we employ causal interventions on the evidence by omission to study when information is (in)sufficient for a model's prediction. Elazar et al. (2021) also use causal interventions that estimate the importance of a property by removing it from a representation. By comparison, even though text-level causal interventions are more intricate due to the discrete nature of text, we perform them on the text itself, by following linguistic rules for optional constituents to preserve the semantics and the fluency of the text. Thorne and Vlachos (2021) perform re-writing of claims by masking and then correcting separate words. They thus generate claims supported by the evidence, particularly for claims not supported before the factual correction. In similar vein, Wright et al. (2022) decompose long, scientific claims into shorter, atomic claims. They then generate negative instances for those by masking single words in claims and replacing them with antonyms retrieved from a scientific knowledge base. In contrast, we perform omissions of evidence information at the sentence and constituent levels and for the new task of Evidence Sufficiency Prediction.

Contrastive Learning (CL) and Counterfactual Data Augmentation (CAD). Most existing work of CL in NLP employs contrastive self-learning for model pre-training (Rethmeier and Augenstein, 2021). Contrary to this, Rethmeier and Augenstein (2022) propose for CL to be performed jointly with the supervised objective. We follow the latter to improve the performance of FC models in detecting when important information is missing from the evidence, by using the original evidence texts paired with evidence texts with omitted information as contrastive data points. We perform contrastive self-training jointly with the supervised objective, as we use the contrastive loss as an unsupervised training for Evidence Sufficiency Prediction. In contrast, using it for pre-training followed by supervised training could lead to the models forgetting the information learned during pre-training, which is needed to improve the performance on *SufficientFacts*. An important factor for CL is the augmentation of negative and positive instances, which can be challenging due to the discrete nature of text. Related work explores augmentation through back-translation (Sennrich et al., 2016), masked word substitution with an LM (Wu et al., 2019), graph neighbourhood sampling (Ostendorff et al., 2022), mix-up (Chen et al., 2020), or a combination thereof (Qu et al., 2021). In a similar vein, automated approaches for CAD in NLP include paraphrasing (Iyyer et al., 2018), and controlled (Madaan et al., 2021) text generation, which do not necessarily change the target label of an instance. CAD is found to improve model robustness to data artifacts (Kaushik et al., 2020; Teney et al., 2020) and to perform better out of domain (Samory et al., 2021). In contrast, we use evidence omission, combined with tri-training for contrastive negative evidence mining (§2.5).

Knowledge-Base Fact Checking. A relevant line of work conducts FC against knowledge bases (KBs) by finding fact triple chains that are (in)consistent with the claim (Kim and Choi, 2021). Discovering such missing triples could also be used to detect insufficient evidence information. As KBs can contain an incomplete set of fact triples, related work completes KBs from unstructured textual data on the Web (Trisedya et al., 2019) or with graph embedding techniques (Kim et al., 2018).

Dataset/Size	Example
FEVER 145,449 train 999,999 dev 999,999 test	**Label**: REFUTES ($\in$ {SUPPORTS, REFUTES, NOT ENOUGH INFO}) **Claim**: Sindh borders Indian states and is in India. **Evidence**: [Sindh] Sindh is home to a large portion of Pakistan's industrial sector and contains two of Pakistan's commercial seaports – Port Bin Qasim and the Karachi Port.
Vitamin C 370,653 train 63,054 dev 55,197 test	**Label**: SUPPORTS ($\in$ {SUPPORTS, REFUTES, NOT ENOUGH INFO}) **Claim**: Westlife sold more than 1 m. video albums and made over 23.5 m. sales in the UK. **Evidence**: [Westlife] According to the British Phonographic Industry (BPI), Westlife has been certified for 13 m. albums, 1.3 m. video albums, and 9.8 m. singles, with a total of more than 24 m. combined sales in the UK.
HoVer 18,171 train 1818 dev 4,000 test	**Label**: NOT SUPPORTED ($\in$ {SUPPORTS, NOT SUPPORTS=(REFUTES+NOT ENOUGH INFO)} **Claim**: Reason Is Treason is the second single release from a British rock band that are not from England. The band known for the early 90's album Novelty are not from England either. **Evidence**: [Kasabian] Kasabian are an English rock band formed in Leicester in 1997. [Jawbox] Jawbox was an American alternative rock band from Washington, D.C., United States. [Reason Is Treason] "Reason Is Treason" is the second single release from British rock band Kasabian. [Novelty (album)] Novelty is an album from the early 90's by Jawbox.

Table 2.1: Sizes and examples instances for the studied fact checking datasets (see §2.3).

This work uses machine learning models that use textual evidence as input instead of performing an intermediate step of completing a knowledge base with needed fact triples.

2.3 Datasets

We employ three fact checking datasets (see Table 2.1) and use the gold evidence documents, i.e., we do not perform document or sentence retrieval (apart from for the ablation experiment in Section 2.6.4). Thus, we avoid potential enforced biases for the veracity prediction models if they had to learn to predict the correct support of the evidence for the claim given wrong evidence sentences. Hence, each of the three fact checking datasets $D = \{(x_i, y_i)|x_i = (c_i, e_i), i \in [1, |D|]\}$ consists of instances with input x_i and veracity labels y_i. The input is comprised of a claim c_i and gold evidence e_i. The veracity label $y_i \in$ {0=SUPPORTS, 1=REFUTES, 2=NEI} for FEVER and VitamiC, and $y_i \in$ {0=SUPPORTING, 1=NOT SUPPORTING} for HoVer.

FEVER (Thorne et al., 2018) contains claim-evidence pairs, where the evidence consists of sentences from Wikipedia pages, and the claims are written manually based on the content of those Wikipedia pages. 87% of the claims have evidence consisting of one sentence. The dataset has a high ratio of token overlap between the claim and the evidence, where the overlap is naturally higher for claims that are supporting (69%), than refuting (59%) and NEI (54%) claims. The high overlap ratio can create a bias for learning from token overlap, which can further prevent generalisation, as also noted in related work (Schuster et al., 2021).

Vitamin C (Schuster et al., 2021) is a collection of sentences from Wikipedia containing factual edits. For each factual edit, annotators construct a claim that is SUPPORTED and one that is REFUTED with the old and the new version of the evidence. When the factual edit introduces/removes facts from the evidence, claims are constructed so that there is NOT ENOUGH INFORMATION (NEI) to support them. Due to its contrastive nature and reduced claim-evidence overlap, the authors demonstrate that models trained on the dataset gain a 10% accuracy improvement on adversarial fact verification.

HoVer (Jiang et al., 2020) is designed to collect claims that need several hops over Wikipedia evidence sentences to verify a claim. The evidence contains between two and four sentences from different Wikipedia articles. As the test dataset is blind and we use the gold evidence, we use the development set for testing purposes and randomly select 10% of the training dataset for development.

2.4 Evidence Omission

To study what types of information the evidence models consider important, we propose to conduct causal interventions for the evidence by omitting information from it. We hypothesise that removing information important for the model to predict the support of evidence for a claim will cause a change in its original prediction, leading to the model indicating that there is missing information. If the removed information is not important for the model though, removing it would not change the model's prediction. We then ask whether the information that is important for a model when predicting the support of the evidence text for a claim, is actually important as judged by human annotators. The human annotations allow for a systematic study of common model errors, i.e., when the models still predict the correct label even if important evidence information has been removed and when they consider the information to be insufficient if unrelated evidence has been removed.

Type	L	Claim	Evidence
S	R	The Endless River is an album by a band formed in 1967.	[[The Endless River]] The Endless River is a studio album by Pink Floyd. [[Pink Floyd]] Pink Floyd were founded in 1965 by students . . .
PP	R	Uranium-235 was discovered by Arthur Jeffrey Dempster in 2005.	[[Uranium-235]] It was discovered in 1935 by Arthur Jeffrey Dempster.
NOUNM	S	Vedam is a drama film.	[[Vedam (film)]] Vedam is a 2010 Indian drama film written and directed by Radhakrishna Jagarlamudi . . .
ADJM	S	Christa McAuliffe taught social studies.	[[Christa McAuliffe]] She took a teaching position as a social studies teacher at Concord High School. . .
ADVM	S	Richard Rutowski heavily revised the screenplay for Natural Born Killers.	[[Natural Born Killers]] The film is based on an original screenplay that was heavily revised by writer David Veloz , associate producer Richard Rutowski . . .
NUMM	S	Being sentenced to federal prison is something that happened to Efraim Diveroli.	[[Efraim Diveroli]] Diveroli was sentenced to four years in federal prison .
DATEM	R	Colombiana was released 1st October 2001.	[[Colombiana]] Colombiana is a French action film from 1st October 2011 . . .
SBAR	R	North Vietnam existed from 1945 to 1978.	[[North Vietnam]] North Vietnam, was a state in Southeast Asia which existed from 1945 to 1976.

Table 2.2: Examples from the FEVER dataset of constituent types (§2.4.1) removed from the evidence for a claim with Label (L) one of SUPPORTS (S) or REFUTES (R).

2.4.1 Evidence Omission Generation

We omit information from the evidence text at the sentence and constituent level. Particularly, we aim to remove information from the evidence such that it does not change its stance towards the claim from SUPPORTS to REFUTES, or vice-versa, while preserving the grammatical correctness and fluency of the evidence. Following studies of linguistic sentence structure (Burton-Roberts, 2016; Börjars and Burridge, 2019), illustrated with examples in Table 2.2, we collect prepositional phrases, modifiers and other optional sentence constructs – i.e. those constructs that can be removed from the sentence without impairing its grammatical correctness, and where the remaining text is semantically identical to the original one, except for the additional information from the removed construct (Garvin, 1958). We use the following optional sentence constructs:

Sentences (S). In FEVER and HoVer, the evidence can consist of more than one sentence. The separate sentences are supposed to contain information important for the fact check, which we further verify with manual annotations as explained in Section 2.4.2. VitaminC consists of single sentences only, and we thus only perform constituent-level omissions for it, as described next.

Prepositional Phrases (PP) are optional phrases that are not part of a Verb Phrase (VP), but are child nodes of the root sentence in the constituent tree (Brown et al., 1991). These usually function as adverbs of place and consist of more than one word.

Noun Modifiers (NOUNM) are optional elements of a phrase or clause structure (Huddleston and Pullum, 2005). NOUNM can be a single or a group of nouns that modify another noun.

Adjective Modifiers (ADJM) are a single or a group of adjectives that modify a noun.

Adverb Modifiers (ADVM) are a single or a group of adverbs that modify verbs, adjectives, or other adverbs and typically express manner, place, time, etc.

Number Modifiers (NUMM) are a single or a group of words denoting cardinality that quantify a noun phrase.

Date Modifiers (DATEM) are a single or a group of words that express temporal reference. To preserve fluency, from a date expression consisting of a day, a month, and a year, we omit either the date, the date and the month, or the year.

Subordinate Clauses (SBAR) are introduced by a subordinating conjunction. Subordinate clauses depend on the main clause and complement its meaning. SBARs can be adverb clauses, adjective clauses, and noun clauses.

For the omission process, we use two pre-trained models with high performance from the Spacy library[1] – a part-of-speech (PoS) tagger with an accuracy of 97.2 and a constituency parser (Kitaev and Klein, 2018) with an F_1-score of 96.3 on the revised WSJ test set (Bies et al., 2015). During the omission process, we use the PoS tags to find nouns, adjectives, adverbs, and numbers and use the constituency tags to select only the modifiers. Thus, we find the NOUNM, ADJM, ADVM, and NUMM constructs. We collect SBAR and PP constructs by finding their corresponding tags in the constituent dependency tree. Finally, for the date, we use two regular expressions that are common date templates used in Wikipedia articles – <month name, date, year> or <date, month name, year>, and remove parts from the templates that

[1] https://spacy.io/

preserve the coherency – <date>, <year>, <month name and date>, or <year and date>.

Overall, in this work, we perform a study of insufficient evidence for FC by removing information from the gold evidence. As explained in Section 2.2, we perform causal interventions on the evidence by omission to study when information is (in)sufficient for a model's prediction. Replacement of words is another operation that can be applied to the evidence. We can, for example, replace different types of named entities with pronouns, and different parts of the speech with demonstrative pronouns to induce insufficient information. However, the replacement operation does not allow for direct causal conclusions as any change of a word with another could potentially lead to confounding factors of the newly introduced word and the model's predictions. Note that, there are some pronouns used in the evidence when they refer to the person/object of the article. We do not treat such cases as insufficient information as the title of the page with the name of the person/object is always prepended to the sentence, which allows for coreference resolution. Finally, another possible operation is the insertion of new information, which would lead to insufficient evidence when performed on the claim. The latter, however, requires the insertion of text that preserves the grammatical correctness and meaning of the claim, which is hard to achieve in an automated way.

2.4.2 Manual Annotations.

Models. We train three Transformer-based FC models – BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020). BERT is pre-trained with masked language modeling and next sentence prediction objectives on the Toronto Book Corpus (Kiros et al., 2015) and the English Wikipedia.[2] It is also the most widely used pre-trained Transformer model.[3] RoBERTa improves upon BERT by optimising key hyper-parameters, and is trained without the next sentence prediction objective. RoBERTa is one of the top-performing models on the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks comprised of various NLP tasks. The latter also holds for ALBERT, another Transformer architecture that improves upon BERT. It does so with parameter-reduction techniques, which lower the memory consumption of the model. ALBERT also employs a self-supervised pre-training loss for inter-sentence coherence. The latter is found to be beneficial for tasks with multiple sentences, and Schuster et al. (2021) report improved FC robustness with it on VitaminC compared to BERT.

[2] https://en.wikipedia.org

[3] https://huggingface.co/models

We train each model on the respective training splits of each dataset with the claim c and the gold evidence e as input to predict the gold veracity label y: $f(c, x) = \hat{y}$. We optimise the supervised cross-entropy loss:

$$\mathcal{L}^S = -\frac{1}{m}\sum_{j=1}^{m} y^j \cdot \log(\hat{y}^j) \tag{2.1}$$

where m is the label space size.

We then use an ensemble of these three different Transformer-based FC models to collect predictions for our new task Evidence Sufficiency Prediction, as we want to find instances with omitted information that are more broadly applicable (e.g., those on which the models agree). The (dis)agreements between the models also allow us to study the differences between them in detecting omitted information. Transformer Language Models are pre-trained on large datasets, the veracity of which can change over time (Schuster et al., 2021). This makes it important that the FC models take into account the facts in the given evidence. When provided with differences and similarities in the three FC models' predictions, future work could then also investigate the degree to which different Transformer-based FC models encode FC-relevant world knowledge they default to in their predictions.

Annotation Task. Next, we collect evidence with removed information as described above. We then use the models to find which of the omitted evidence they consider important, resulting in a prediction change to NEI. We consider instances from the original test splits of each of the datasets, where all models predicted the veracity correctly before the evidence omission was performed, as these are the cases where we can observe whether evidence omission causes the veracity prediction to change to NEI. We collect instances with omitted evidence information where the models: (1) agree that the evidence is still enough vs. (2) insufficient; and where they (3) disagree in their prediction. We collect a total of 400 instances at the sentence, and 600 instances at the constituent level from the test splits of the corresponding datasets, distributed equally among the above three groups.

We employ annotators on Amazon Mechanical Turk[4]. We first train potential annotators, presenting them with annotation guidelines and illustrative examples. We then select annotators using a qualification test with nine test annotations for our task. Each annotation had the cost of 0.10$, and annotators were paid 10$ on average per hour. The annotation task is to determine whether the evidence is still sufficient for predicting the label without the omitted information. If the remaining evidence is still sufficient, we ask them for the reason – whether this is because the removed evidence

[4] https://www.mturk.com/

is repeated in the remaining text or because the removed evidence is not relevant to the veracity of the claim. Following the annotation guidelines for FEVER and HoVer, we ask the annotators not to use any world knowledge or knowledge they might have about the claim. For more details on the annotation task and the guidelines, we will release the dataset with a detailed README file.

The final dataset *SufficientFacts* $= \{(x'_i, y'_i) | x'_i = (c_i, e'_i), i \in [1, |\textit{SufficientFacts}|]\}$ consists of test instances x'_i with labels y'_i. All of the instances in *SufficientFacts* are a subset of the instances in the test datasets of FEVER, VitaminC, and HoVer with the following changes. The input x'_i is comprised of the original claim c_i and the evidence with omitted information e'_i. The tokens of e'_i are a subset of the tokens of the original gold evidence e_i of the instance. To re-iterate, the label of the originally selected instances is either SUPPORTS or REFUTES, i.e. they have sufficient gold evidence information, where after omitting information from the evidence, the new label y'_i becomes either NEI if the majority of the annotators selected that important information was removed, and otherwise remains the original label – SUPPORTS and REFUTES for FEVER and VitamiC, or SUPPORTING for HoVer.

The resulting inter-annotator agreement (IAA) for *SufficientFacts* is 0.81 Fleiss' κ from three annotators. Due to the novelty of the introduced task of Evidence Sufficiency Prediction, we do not have direct points of comparison for IAA. However, we point as a reference the IAA reported for the related task of fact checking for the HoVer dataset – 0.63 Fleiss' κ, and for the FEVER dataset – 0.68 Fleiss' κ, where, for both datasets, the annotators were thoroughly trained and highly paid. The biggest challenges for our annotators, judging by their errors during the qualification test, were not to use common knowledge and assumptions in their annotations, and the general complexity of the task.

2.4.3 *SufficientFacts* Analysis.

Overall Agreement with Annotators. The statistics of the resulting dataset, *SufficientFacts*, are presented in Table 2.3. We find that all three models agree that the remaining evidence is still sufficient (EI Agree) even when it has become insufficient after omitting information needed for verifying the claim (NEI) in 430 out of 1000 instances. We assume that these failures of all three models to detect missing information for FC point to the models making predictions based only on patterns observed in claims, or to the models defaulting to world knowledge encoded in the pre-trained Transformer models. We further find that when the models disagree about whether the remaining information is still sufficient (Disagree), they disagree mostly about instances where the omitted evidence information is needed for veracity prediction (NEI)

Dataset	Model Pred	EI_I	EI_R	NEI
FEVER SENT	EI Agree	61	20	119
	NEI Agree	13	9	178
	Disagree	39	24	137
	Total	113	53	434
FEVER CONST	EI Agree	146	3	51
	NEI Agree	0	0	200
	Disagree	43	1	156
	Total	189	4	407
HoVer SENT	EI Agree	32	12	156
	NEI Agree	4	1	195
	Disagree	7	1	192
	Total	43	14	543
HoVer CONST	EI Agree	139	6	55
	NEI Agree	1	0	199
	Disagree	48	1	151
	Total	188	7	405
VitaminC CONST	EI Agree	**146**	**5**	49
	NEI Agree	0	0	**200**
	Disagree	13	0	187
	Total	159	5	436
Total	EI Agree	524	46	430
	NEI Agree	18	10	972
	Disagree	150	27	823
	Total	692	83	2225

Table 2.3: Statistics of *SufficientFacts* presenting the predictions of the models in the ensemble (Model Pred: Agree Enough Information (EI Agree), Agree Not Enough Information (NEI Agree), Disagree, and Total) vs human annotations of the same (EI – Irrelevant (EI_I), EI – Repeated (EI_R), NEI). We present sentence (SENT) and constituent omission (CONST) dataset splits separately. We embolden/underline results of the datasets for predictions where the three models agree (NEI Agree, EI Agree) and have the highest/lowest agreement with human annotations about EI_I, EI_R and NEI predictions. We use light blue/dark blue to denote where lower/higher results are better.

– in 823 out of 1000 instances. By contrast, when the models agree that the remaining evidence is insufficient, they are correct in 972 out of 1000 of the instances.

Separate Dataset Agreement with Annotators. Looking at the separate datasets, it is the hardest for the models to identify missing evidence information needed for the fact check (EI Agree vs. NEI) for HoVer, particularly with sentence omissions, and the easiest for the VitaminC dataset with constituent omissions. We hypothesise that the latter is due to the HoVer dataset having more complex claims and requiring cross-sentence reasoning, whereas VitaminC contains contrastive instances which, during training, guide the models to identify the parts of the evidence needed for FC. Overall, the models fail to detect missing information more from sentences rather than from constituents. We hypothesise that this effect can be observed partly because models struggle to conduct multi-hop reasoning over them. Another possible reason

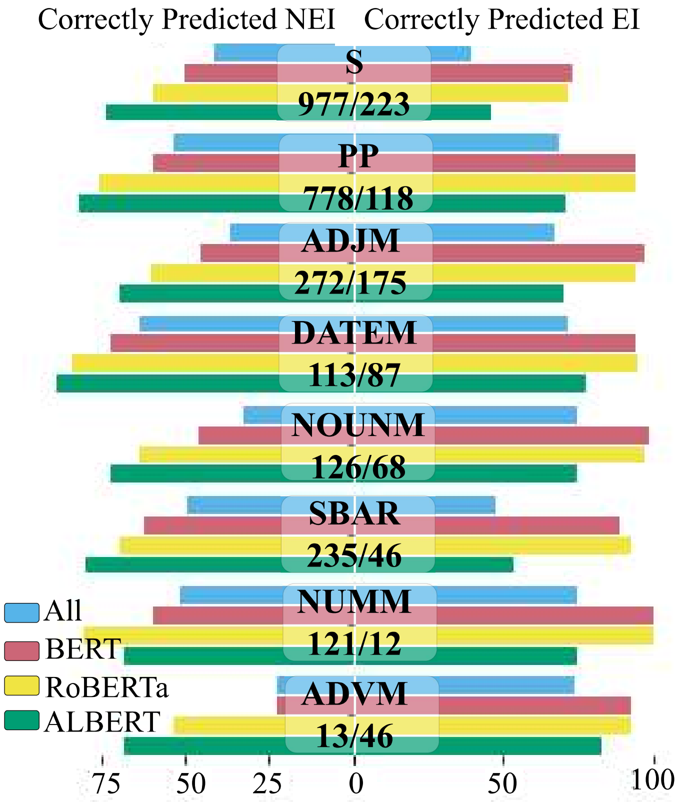
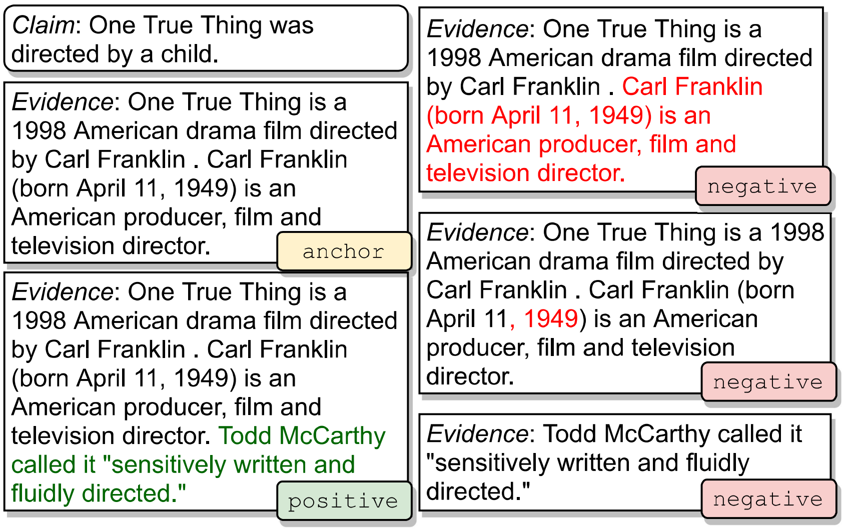


Figure 2.2: *SufficientFacts* – fine-grained analysis by type of removed evidence inftype (§2.4.1) vs. proportion of correct predictions of NEI/EI instances. The proportion is computed for the separate models – BERT, RoBERTa, ALBERT, and for all three models agreeing on the correct NEI/EI label (All). The total number of NEI/EI instances of each type is provided under each of the types of removed evidence information. *A higher* proportion of correct predictions is *better*.

for that is that the models could be better at verifying the type of information removed from a sentence constituent rather than from a sentence.

Performance by Omitted Evidence Type and Model. Figure 2.2 provides a fine-grained analysis of the performance of the models for different types of omitted constituents. We observe that it is the hardest to detect when the evidence is missing information for the prediction (Correctly Predicted NEI) that was removed from adverbial modifiers (ADVM), followed by subordinate clauses (SBAR). By contrast, it is easiest to detect missing information when it is a date modifier (DATEM), followed by number modifiers (NUMM). BERT has the lowest rate of correctly detecting insufficient evidence from the three models, followed by RoBERTa, whereas ALBERT performs best. We conjecture that this is due to RoBERTa being an optimisation of BERT, and due to ALBERT including pre-training with an inter-sentence coherence objective, which has been shown to make the model more robust for factual verification (Schuster et al., 2021). Even though ALBERT contains fewer parameters than BERT, it still



Claim: One True Thing was directed by a child.

Evidence: One True Thing is a 1998 American drama film directed by Carl Franklin . Carl Franklin (born April 11, 1949) is an American producer, film and television director.

Evidence: One True Thing is a 1998 American drama film directed by Carl Franklin . Carl Franklin (born April 11, 1949) is an American producer, film and television director. Todd McCarthy called it "sensitively written and fluidly directed."

Evidence: One True Thing is a 1998 American drama film directed by Carl Franklin . Carl Franklin (born April 11, 1949) is an American producer, film and television director.

Evidence: One True Thing is a 1998 American drama film directed by Carl Franklin . Carl Franklin (born April 11, 1949) is an American producer, film and television director.

Evidence: Todd McCarthy called it "sensitively written and fluidly directed."

Figure 2.3: Example of augmented contrastive instances for the original (anchor) instance. Red designates removed evidence information, where the models agree that the remaining evidence is not sufficient, producing a negative contrastive instance. Green designates an added distractor sentence, producing a positive instance. The distractor sentence, selected to have high overlap with the claim but with insufficient information, is used as another negative instance.

detects better when the evidence is insufficient. Finally, we see a natural trade-off between correctly detecting sufficient and correctly detecting insufficient information. In particular, some models such as ALBERT have a higher number of correct predictions on instances without enough information (Fig. 2.2, left). However, on instances with sufficient evidence information (Fig. 2.2, right), ALBERT has the lowest number of correct predictions. In contrast, BERT has the worst performance on the NEI instances, but the best performance on EI instances.

2.5 Evidence Omission Detection

To improve the performance of models in recognising when the evidence is not enough for verifying a claim, we experiment with CAD (§2.5.2) and a CL loss (§2.5.1). Both methods use contrastive data augmented with the proposed evidence omission method (§2.4.1) in combination with tri-training, as illustrated in Fig. 2.3. We omit information from the original (anchor) evidence to collect potential negative instances with missing important evidence information compared to the original evidence (Fig. 2.3, right). From the resulting candidates, we select as negative only those predicted as having insufficient information by the other two supervised models from the ensemble (§2.4) (e.g., RoBERTa and ALBERT predict NEI when we are training a model with a BERT Transformer architecture). We also collect positive instances that still have sufficient evidence information after applying a data augmentation operation. For each instance x_i, we find one distractor sentence from the document of the gold evidence that is the most similar to the claim by word overlap. We append the distractor sentence to the original evidence, which serves as a positive instance (Fig. 2.3, left). Finally, we include only the distractor sentence as a negative instance as it does not have enough evidence contrasted both with the positive and the anchor instances. We conjecture that the latter would serve as a training signal for avoiding the bias for overlap between the claim and the evidence.

2.5.1 Contrastive Learning

We study self-supervised learning to train FC models that recognise when the evidence is not enough for verifying a claim. In particular, we propose to use self-supervised contrastive learning (CL) jointly with the supervised learning of the model to predict the support of the evidence for a claim. Given an anchor instance x_i, a positive instance x_i^+, and K^- negative instances $x_{i,k}^-$, $k \in [1, K^-]$, the objective of CL is to make the anchor and the positive instance closer in the representation space, and the anchor and the negative instances further apart. The anchor, positive, and negative instances are collected and/or augmented from the training splits of the corresponding datasets as described above. Each model, $g(x) = l(h(x)) = l(e) = \hat{y}$, uses 12 encoding layers to encode an input instance $h(x) = e$ and uses the encoding e of the last encoding layer to predict the veracity label with a linear layer: $l(e) = \hat{y}$. We encode the anchor, the positive, and the negative instances with the corresponding model g, resulting in the anchor e_i, the positive e_i^+, and the negative $e_{i,j}^-$ representations, and minimise the following CL loss:

$$\mathcal{L}^{\text{CL}} = \log \sigma(s(e_i, e_i^+;\tau) + \sum_{k=1}^{K^-} log \sigma(1 - s(e_i, e_{i,k}^-;\tau)) \tag{2.2}$$

where s is a similarity function between the representation of the two instances – cosine similarity in our case, τ is a temperature parameter subtracted from the cosine similarity (Ma and Collins, 2018), and K^- is the number of negatives. Note that the CL loss is the same as Noise Contrastive Estimation (Ma and Collins, 2018) expressed as a binary objective loss. The representation of each instance is obtained by mean pooling of the word representations of the instance in the last layer of the model M. We include the contrastive self-learning loss for those instances that are not annotated as NEI, as we cannot construct contrastive negative evidence with insufficient information for the instances that already do not have enough information for verification. Finally, the CL loss is optimised jointly with the supervised loss:

$$\mathcal{L}^S = -\frac{1}{m}\sum_{j=1}^{m} y^j \cdot \log(\hat{y}^j) \tag{2.3}$$

$$\mathcal{L} = \mathcal{L}^S + \mathcal{L}^{\text{CL}} \tag{2.4}$$

where $\hat{y}_i$ is the label prediction of model M, m the label space size, y_i is the gold label for instance x_i, $y_i \in$ {0=SUPPORTS, 1=REFUTES, 2=NEI} for FEVER and VitamiC, and $y_i \in$ {0=SUPPORTING, 1=NOT SUPPORTING} for HoVer.

2.5.2 Counterfactual Data Augmentation

We also experiment with counterfactually augmented evidence, using the negative and positive instances constructed as described above (§2.5 and Fig. 2.3). As the models have high accuracy when they agree that a piece of evidence with omitted information is not sufficient (see agreement with human annotations in Table 2.3), we conjecture that the counterfactually augmented instances would serve as a good training signal for detecting (in)sufficient evidence information without incurring annotation costs for training data. The counterfactually augmented data is thus simply combined with the training instances of each dataset. In particular, we include in the training set the claim and the original evidence (anchor) with the corresponding gold label y_i. We include the positive instance – original evidence with distractor sentence appended to it, with the original gold label y_i. The negative instances, i.e., with insufficient evidence information, are included with a gold label y_i = NEI for FEVER and VitaminC, and y_i = NOT SUPPORTING for HoVer. Each model, $h(c, e) = \hat{y}$, receives as input the original claim c and the augmented or the original evidence e and predicts the veracity label $\hat{y}$. We optimise a supervised cross-entropy loss as per Equation 2.3.

2.5.3 Baseline Ensemble

We include a simple ensemble, consisting of the three models – BERT, RoBERTa, and ALBERT. Each ensemble contains only supervised models (§2.4.2), models trained with CAD (§2.5.2), or models trained with CL loss (§2.5.1). We employ majority voting, where the final prediction is the most common class among the predictions of the three models on an instance, defaulting to the class with the highest predicted probability if there is no most common class.

Dataset	Model	Veracity Pred. / Orig.Test				Evidence Sufficiency / Suff.Facts			
		BERT	RoBERTa	ALBERT	Ens.	BERT	RoBERTa	ALBERT	Ens.
FEVER	Supervised	87.16	88.69	86.67	88.81	59.51	59.10	63.00	61.36
	+ CL	87.62	88.81	86.62	89.02	65.79	67.98	**70.83**	**69.90**
	+ CAD	**87.86**	**89.23**	**87.31**	**89.14**	**67.18**	**69.58**	68.56	69.25
HoVer	Supervised	80.75	83.37	76.88	82.73	58.15	64.81	66.28	65.88
	+ CL	81.82	83.38	77.62	83.08	74.91	75.41	72.83	78.05
	+ CAD	**81.87**	**83.65**	**79.44**	**83.65**	**74.98**	**77.14**	**76.12**	**79.07**
VitaminC	Supervised	82.26	84.98	83.38	86.01	58.51	69.07	66.57	66.76
	+ CL	83.00	**85.54**	83.48	**86.22**	62.34	72.18	68.13	70.42
	+ CAD	**83.56**	85.65	**83.82**	86.14	**72.93**	**75.79**	**75.13**	**78.60**

Table 2.4: Macro F_1-score test performance of models and an ensemble (Ens.) (§2.5.3) trained on the supervised training splits of each dataset (Supervised), and in addition with the contrastive objective (+CL) (§2.5.1) and the counterfactually augmented data (+CAD) (§2.5.2). Results are the average of three different seed runs. The highest results for a test dataset and a model are in bold, and the overall highest result of a model for a test dataset are additionally underlined.

2.5.4 Experimental Details

All models are trained on the respective training splits of each dataset. We select the checkpoint with the highest macro F_1-score on the dev sets and provide results on the test sets. We note that for the newly introduced task Evidence Sufficiency Prediction, we have an annotated test dataset *SufficientFacts*, but no training dataset. The training is performed on the original training splits of the corresponding datasets, which have a different label distribution from the introduced diagnostic test set. Hence, it is possible that some of the instances in *SufficientFacts* are out of the original training distribution, which would make this diagnostic dataset of rather adversarial nature.

We select the learning rate = $1e-5$ and the temperature parameters $\tau = 1.5$ by grid search over the performance on the dev sets from $[1e\text{–}5, 2e\text{–}5, 3e\text{–}5]$ and $[0, 0.5, 1, 1.5, 2]$ respectively. We use the batch sizes for corresponding models from prior work – 8 for HoVeR, 32 for FEVER, and 16 for VitaminC.

2.6 Results and Discussion

2.6.1 Supervised Model Performance

We start by discussing the performance of models trained on the supervised splits of the corresponding datasets to predict labels for claims based on the newly created dataset *SufficientFacts* for Evidence Sufficiency Prediction, presented in Table 2.4. Recall that the instances in *SufficientFacts* had correct predictions from all models before the evidence omission was performed (§2.4.2), i.e., the performance of the models on the instances in *SufficientFacts* had 100 F_1-score before the evidence omission. Hence, the omission of information from the evidence results in a performance decrease from 100 to 58 F_1-score (BERT model for the HoVer dataset), i.e. a decrease of up to 42 F_1-score. Out of the three FC models, BERT has the lowest performance on *SufficientFacts*, whereas ALBERT has the highest. The latter corroborates that ALBERT is a more robust model for fact verification, as explained in more detail in Section 2.4.2.

Further, we observe the worst performance on *SufficientFacts* for the HoVer dataset – down to 58 F_1-score, followed by FEVER, and with the best performance on VitaminC. We suggest that the contrastive nature of the instances in VitaminC that contain factual edits of the evidence, changing the support of the evidence for the claim, as described in Section 2.3, can indeed provide a better learning signal for the models about which parts of the evidence are important for verifying the claim.

2.6.2 Contrastive Loss and Augmented Model Performance

Including a CL loss or CAD results in improvements for all models and datasets on *SufficientFacts* by up to 17.2 F_1-score. Note that the proposed technique does not incur additional annotation costs for training data for Evidence Sufficiency Prediction. This corroborates that our proposed evidence omission approach combined with tri-training improves the recognition of (in)sufficient evidence. This, in turn, improves the performance on the original test sets by up to 3.6 F_1-score. Comparing the CL loss with counterfactually augmented data, we see that CAD improves the model performance in more cases on *SufficientFacts*, except for ALBERT for the FEVER dataset. This could be because the augmented data uses raw labels obtained with tri-learning, while the CL loss only drives apart the negative instances from the anchor in the representation space.

Finally, we compare the performance of CAD and CL loss that rely on the agreement predictions of the supervised models with the simple majority voting ensembles (§2.5.3). Single models trained with CAD and CL loss still outperform the ensembles of the supervised models. A majority voting classifier from the models trained with CAD and CL loss improves the performance on the original and *SufficientFacts* sets even further.

2.6.3 Comparison to Related Work

We further compare the performance of our models to existing systems on the used datasets (see Table 2.5). Note that we are particularly interested in veracity prediction to study what evidence models consider as sufficient for factuality prediction. Thus, in the base setting, we do not conduct evidence retrieval, as typically performed for

Dataset	Model	F_1
FEVER	DA *(Thorne et al., 2018)*	83.84
	RoBERTa Supervised	88.69
	+ CL	88.68
	+ Augmented	**89.23**
HoVer	BERT *(Jiang et al., 2020)*	*81.20*
	BERT Supervised	80.75
	+ CL	81.82
	+ Augmented	**81.87**
VitaminC	ALBERT *(Schuster et al., 2021)*	82.76
	ALBERT Supervised	83.38
	+ CL	83.48
	+ Augmented	**83.82**

Table 2.5: Macro F_1-score on the original test set compared to baseline (FEVER) and SOTA (HoVer, VitaminC) oracle results. Highest results for a dataset are in bold.

the HoVer and FEVER datasets, but train models using gold evidence (oracle). For FEVER, existing systems report results on both tasks, hence we can only compare to the veracity prediction results with oracle evidence available in the FEVER dataset paper with a Decomposable Attention (DA) model (Parikh et al., 2016). For HoVer and VitaminC, the presented results are also from the dataset papers of models trained with oracle evidence. As there are no other reported results on these datasets, they also represent the state-of-the-art for these two datasets. To compare to them, we pick those of our models with the same Transformer architecture as used in the respective dataset papers, and the best-performing model architecture for FEVER. Note that we use the same training setting as in related work (§2.5.4) for all models and datasets. We find that our supervised models are close in performance to prior reported results. Furthermore, including counterfactual data augmentation and contrastive learning leads to improvements over prior results for all three datasets, by up to 2.6 F_1-score.

2.6.4 Incorrect Evidence

So far, we studied model performance on instances with omitted information from the gold evidence. We now probe how well the models detect missing information given retrieved incorrect evidence, which does not contain sufficient information. The latter is possible in real-world scenarios. The evidence we feed to the fact checking model depends on the preceding evidence retrieval step, which can retrieve gold evidence with varying performance. While the fact checking model is possibly trained on gold evidence to avoid learning spurious correlations, we want to evaluate its capability to recognise when the retrieval system has discovered incorrect evidence as

Model	BERT	RoBERTa	ALBERT	Ens.
FEVER				
Supervised	82.18	81.88	85.03	84.24
+ CL	87.63	93.53	**95.18**	**91.60**
+ CAD	**89.50**	**94.73**	90.89	90.95
HoVer				
Supervised	97.27	78.64	97.65	88.57
+ CL	99.58	**99.71**	**99.45**	**99.98**
+ CAD	**99.65**	98.52	99.30	99.97
VitaminC				
Supervised	69.99	80.36	**80.69**	78.33
+ CL	75.77	79.32	78.95	78.90
+ CAD	**80.71**	**82.69**	75.69	**80.78**

Table 2.6: Accuracy of models trained on the supervised training splits of each dataset (Supervised), the contrastive objective in addition to training with Supervised (+CL), and the counterfactually augmented data (+CAD). The models are evaluated on the task of Evidence Sufficiency Prediction on datasets with extracted unrelated evidence information (§2.6.4).

well. Note that current FC benchmarks do not consider the prediction of a veracity model if the correct evidence is not retrieved. However, in realistic situations, we do not know whether the evidence is correct, and FC models would still provide a veracity for a claim. Hence, we further study the performance of models on incorrect evidence. For each instance in the original test splits, we retrieve incorrect evidence by selecting the closest evidence of another claim in the dataset by word overlap between the claim and the evidence candidates. We then use the retrieved instead of the original evidence. This results in a test set of claims with incorrect evidence of the same size as the original test split.

Table 2.6 reports results on the test datasets incorrect evidence. As all instances in the dataset have the new gold label of NEI, we report accuracy, which corresponds to the ratio of the instances with a predicted NEI label. We find that the performance of the models is improved by as much as 27 accuracy points after training with CAD or CL, which is another indication for the effectiveness of the proposed training methods. We also find that CAD again brings larger performance gains than CL, except for HoVer, where the two approaches achieve very similar accuracy scores.

The extended evaluation of incorrect evidence is an important complement to the study of missing evidence. However, the two are not necessarily directly comparable. First, in Table 2.4, the two test datasets – the Original Test and SufficientFacts, both have instances with and without sufficient evidence. The extended study on incorrect evidence in this section only has instances that do not have sufficient evidence. This also results in our use of different measures to report results – accuracy in Table 2.6, which is the percentage of detected incorrectly retrieved evidence, and macro F_1 score in Table 2.4, which combines the performance on up to three classes in a balanced way.

However, it is worth addressing the high performance of the models on the irrelevant evidence dataset. We employ evidence that has word overlap with the claim, but is not necessarily semantically similar to the claim. If the models were to only rely on features of the claim or on surface word overlap between the claim and the evidence, the models would have low performance on the irrelevant evidence dataset. We train models to avoid such spurious correlations with CAD and CL loss, which make discovering missing evidence information in irrelevant evidence easy, leading to the observed high performance in Table 2.6.

2.6.5 Error Analysis

Lastly, we conduct an error analysis on the newly introduced *SufficientFacts* to understand whether known biases in models trained on FC datasets (§2.2) also affect predictions on *SufficientFacts*.

Claim-Only Prediction. Schuster et al. (2019) found that FC models often learn spurious correlations and can predict the correct label even when no evidence is provided, as they learn only features of the claim. We investigate whether it is also among the reasons for incorrect predictions of the models on the *SufficientFacts* dataset. We compute the percentage of instances in *SufficientFacts* where the models do not predict when provided with evidence. We find that for the HoVer dataset, the supervised BERT model does not predict an NEI label for 36% of the instances in *SufficientFacts* whereas the respective number for RoBERTa is 23% and 14% for ALBERT. This indicates that supervised models trained on HoVer learn claim-only features for some instances. After training the models with CAD (§2.5.2) and CL loss (§2.5.1), fewer than 1% of instances from *SufficientFacts* are predicted as having enough information by each of thee models when given only the claim. This indicates that training with CAD and CL loss decreases the claim-only bias for the HoVer dataset. For FEVER and VitaminC, we find a lower percentage of instances (fewer than 4%) in the corresponding *SufficientFacts* splits that the supervised models predict as having enough information when given only the claim. We hypothesises that this is due to the larger amount of training data in both datasets and due to the contrastive nature of VitaminC, which requires the models to learn features from the evidence as well. The percentage is again decreased after training with CAD and CL (fewer than 1%). Finally, we find that the instances that are still not detected as having insufficient

1. *Claim:* Unison (Celine Dion album) was originally released by Atlantic Records. *Evidence:* [Unison (Celine Dion album)] The album was originally released on 2 April 1990. *Dataset:* FEVER, *Model:* BERT *Gold:* NEI, *Sup.:* SUPPORTS, +*CAD:* NEI, +*CL:* NEI
2. *Claim:* Jean-Jacques Dessalines was born on October 2nd, 2017. *Evidence:* [Jean-Jacques Dessalines] He defeated a French army at the Battle of Vertières. *Dataset:* FEVER, *Model:* RoBERTa, *Gold:* NEI, *Sup.:* SUPPORTS, +*CAD:* NEI, +*CL:* SUPPORTS
3. *Claim:* The Times is a website. *Evidence:* N/A *Dataset:* FEVER, *Model:* RoBERTa, *Gold:* NEI, *Sup.:*REFUTES, +*CAD:* REFUTES, +*CL:* REFUTES
4. *Claim:* The Bragg–Gray cavity theory was developed by Louis Harold Gray, William Lawrence Bragg, and a man knighted in the year 1920. *Evidence:* [William Henry Bragg] He was knighted in 1920. *Dataset: HoVer, Model : RoBERTa, Gold: NEI, supervised: SUPPORTS, +CAD: SUPPORTS, +CL: SUPPORTS*

Table 2.7: Example model predictions before (Sup.) and after including CAD/CL loss training.

evidence after training with CAD/CL loss are those that the model could have gained world knowledge about during pre-training. One example of such a claim is given in Table 2.7, row 3.

Claim-Evidence Overlap. Schuster et al. (2021) also find that FC models are biased in predicting the SUPPORT class when the overlap between the claim and the evidence is high. We conjecture that this is another possible reason that the instances in *SufficientFacts* are hard for the models to distinguish as having missing important evidence information as their evidence still has a high overlap with the claim. To probe this, we compute the average overlap between the claim and the evidence, disregarding stop words, of instances in the *SufficientFacts* that are predicted as having insufficient information by the supervised models and by the models trained with CAD and CL loss. For FEVER and HoVer, the instances predicted as NEI by the supervised models have low overlap with the claim that increases after training with CAD and CL loss (61% to 68% for HoVer and 63% to 65% for FEVER). An example instance where the evidence has high overlap with the claim and is predicted as NEI only after training with CAD and CL loss can be found in Table 2.7, row 1. The latter is an indication that training with CAD and CL loss also reduces the overlap bias of FC models. We do not observe a change in the overlap ratio for VitaminC, where we assume that training with contrastive instances already prevents learning biases, including the overlap bias.

Spurious Patterns. Finally, we investigate whether the models learn other spurious patterns that could lead to low results on *SufficientFacts*. We already observed that for some instances, the supervised models predict that the evidence is not sufficient after removing irrelevant information (Table 2.3), which is one indication of learned spurious patterns. Further, when removing important information, the supervised models still predict the same label for some instances, as they rely on other parts of the input, which might not be important. Table 2.7 shows one example where the supervised models did not recognise that the evidence is missing important information (row 1), but after training with CAD or CL loss, it was detected as NEI. However, there are still possible spurious correlations that the models learn even after training with CAD or CL loss, e.g. the example in row 4. Another such example is in row 3, where even after training with CAD and CL loss, the models still find the claim without any provided evidence sufficient for predicting a refuted claim. As this example relies on knowledge of common facts, we assume that the models rely on knowledge obtained during pre-training or fine-tuning instead. Finally, we find that CAD can prevent the model from learning spurious correlations more than the CL loss. This leads to more instances having the correct prediction only after training with CAD, as in the example in row 2.

2.7 Conclusion

We propose a new task related to fact checking, namely detecting when evidence with omitted information is (in)sufficient. To this end, we conducted an in-depth empirical analysis with a newly introduced fluency-preserving method for omitting evidence information. We compared what Transformer-based models and humans find to be sufficient information for FC, resulting in a novel dataset, *SufficientFacts*. Finally, we showed that the proposed evidence omission method can be used for collecting contrastive examples for CL and CAD, which improved the performance of the studied models on the Evidence Sufficiency Prediction task and on veracity prediction.

The resulting models could be applied to detect emergent false claims, which gain popularity before any reputable source can refute them, as our proposed models can indicate when the provided input is insufficient for making a decision and whether to provide the user with the veracity prediction. Such models could also be used for detecting knowledge or evidence gaps that need to be filled to refute or support popular claims. Another possible future research direction would be to build FC models that indicate the particular part of the claim that they are missing supporting evidence for. Moreover, our proposed analysis and methods could be applied to other knowledge-intensive tasks, such as question answering.

Acknowledgments

The research documented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199. Isabelle Augenstein's research is further partially funded by a DFF Sapere Aude research leader grant. The authors would like to thank the anonymous reviewers and action editors for their helpful comments and suggestions.

Generating Label Cohesive and Well-Formed Adversarial Claims

3

3.1 Introduction

Adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013) are deceptive model inputs designed to mislead an ML system into making the wrong prediction. They expose regions of the input space that are outside the training data distribution where the model is unstable. It is important to reveal such vulnerabilities and correct for them, especially for tasks such as fact checking (FC).

In this paper, we explore the vulnerabilities of FC models trained on the FEVER dataset (Thorne et al., 2018), where the inference between a claim and evidence text is predicted. We particularly construct *universal adversarial triggers* (Wallace et al., 2019a) – single n-grams appended to the input text that can shift the prediction of a model from a source class to a target one. Such adversarial examples are of particular concern, as they can apply to a large number of input instances.

However, we find that the triggers also change the meaning of the claim such that the true label is in fact the target class. For example, when attacking a claim-evidence pair with a 'SUPPORTS' label, a common unigram found to be a universal trigger when switching the label to 'REFUTES' is 'none'. Prepending this token to the claim drastically changes the meaning of the claim such that the new claim is in fact a valid 'REFUTES' claim as opposed to an adversarial 'SUPPORTS' claim. Furthermore, we find adversarial examples constructed in this way to be nonsensical, as a new token is simply being attached to an existing claim.

Our **contributions** are as follows. We *preserve the meaning* of the source text and *improve the semantic validity* of universal adversarial triggers to automatically construct more potent adversarial examples. This is accomplished via: 1) a *novel extension to the HotFlip attack* (Ebrahimi et al., 2018), where we jointly minimize the target class loss of a FC model and the attacked class loss of a natural language inference model; 2) a *conditional language model* trained using GPT-2 (Radford et al., 2019), which takes trigger tokens and a piece of evidence, and generates a semantically coherent new claim containing at least one trigger. The resulting triggers maintain potency against a FC model while preserving the original claim label. Moreover, the conditional language model produces semantically coherent adversarial examples

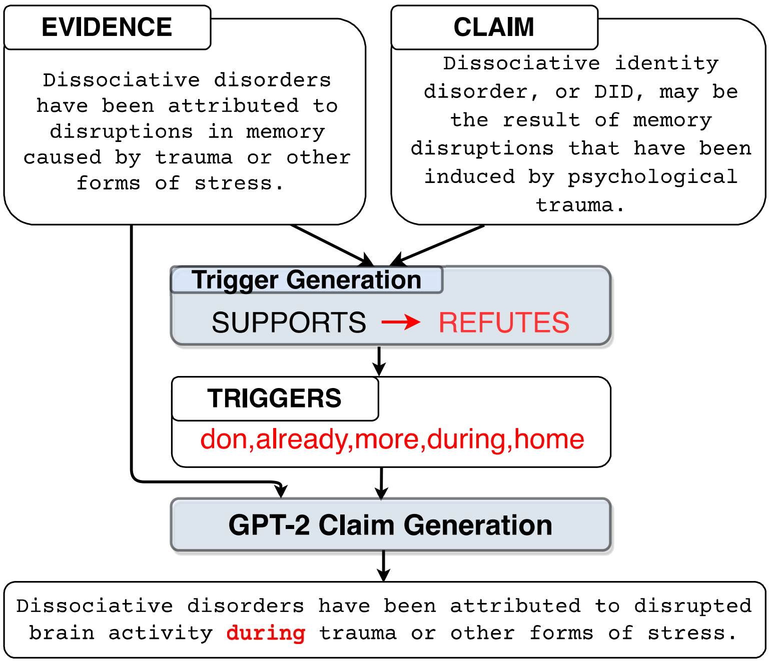

EVIDENCE
Dissociative disorders have been attributed to disruptions in memory caused by trauma or other forms of stress.
CLAIM
Dissociative identity disorder, or DID, may be the result of memory disruptions that have been induced by psychological trauma.
Trigger Generation
SUPPORTS → REFUTES
TRIGGERS
don,already,more,during,home
GPT-2 Claim Generation
Dissociative disorders have been attributed to disrupted brain activity during trauma or other forms of stress.

Figure 3.1: High level overview of our method. First, universal triggers are discovered for flipping a source to a target label (e.g. SUPPORTS → REFUTES). These triggers are then used to condition the GPT-2 language model to generate novel claims with the original label, including at least one of the found triggers.

containing triggers, on which a FC model performs 23.8% worse than with the original FEVER claims. The code for the paper is publicly available.[1]

3.2 Related Work

3.2.1 Adversarial Examples

Adversarial examples for NLP systems can be constructed as automatically generated text (Ren et al., 2019) or perturbations of existing input instances (Jin et al., 2019; Ebrahimi et al., 2018). For a detailed literature overview, see Zhang et al. (2020b).

One potent type of adversarial techniques are universal adversarial attacks (Gao and Oates, 2019; Wallace et al., 2019a) – single perturbation changes that can be applied to a large number of input instances and that cause significant performance decreases of the model under attack. Wallace et al. (2019a) find universal adversarial triggers that can change the prediction of the model using the HotFlip algorithm (Ebrahimi et al., 2018).

However, for NLI tasks, they also change the meaning of the instance they are appended to, and the prediction of the model remains correct. Michel et al. (2019) address this by exploring only perturbed instances in the neighborhood of the original

[1] https://github.com/copenlu/fever-adversarial-attacks

one. Their approach is for instance-dependent attacks, whereas we suggest finding *universal* adversarial triggers that also preserve the original meaning of input instances. Another approach to this are rule-based perturbations of the input (Ribeiro et al., 2018) or imposing adversarial constraints on the produced perturbations (Dia et al., 2019). By contrast, we extend the HotFlip method by including an auxiliary Semantic Textual Similarity (STS) objective. We additionally use the extracted universal adversarial triggers to generate adversarial examples with low perplexity.

3.2.2 Fact Checking

Fact checking systems consist of components to identify check-worthy claims (Atanasova et al., 2018; Hansen et al., 2019; Wright and Augenstein, 2020), retrieve and rank evidence documents (Yin and Roth, 2018; Allein et al., 2021), determine the relationship between claims and evidence documents (Bowman et al., 2015; Augenstein et al., 2016; Baly et al., 2018), and finally predict the claims' veracity (Thorne et al., 2018; Augenstein et al., 2019). As this is a relatively involved task, models easily overfit to shallow textual patterns, necessitating the need for adversarial examples to evaluate the limits of their performance.

Thorne et al. (2019a) are the first to propose hand-crafted adversarial attacks. They follow up on this with the FEVER 2.0 task (Thorne et al., 2019b), where participants design adversarial attacks for existing FC systems. The first two winning systems (Niewinski et al., 2019; Hidey et al., 2020) produce claims requiring multi-hop reasoning, which has been shown to be challenging for fact checking models (Ostrowski et al., 2021). The other remaining system (Kim and Allan, 2019) generates adversarial attacks manually. We instead find universal adversarial attacks that can be applied to most existing inputs while markedly decreasing fact checking performance. Niewinski et al. (2019) additionally feed a pre-trained GPT-2 model with the target label of the instance along with the text for conditional adversarial claim generation. Conditional language generation has also been employed by Keskar et al. (2019) to control the style, content, and the task-specific behavior of a Transformer.

3.3 Methods

3.3.1 Models

We take a RoBERTa (Liu et al., 2019) model pretrained with a LM objective and fine-tune it to classify claim-evidence pairs from the FEVER dataset as SUPPORTS, REFUTES, and NOT ENOUGH INFO (NEI). The evidence used is the gold evidence, available for the SUPPORTS and REFUTES classes. For NEI claims, we use the system of Malon (2018) to retrieve evidence sentences. To measure the semantic similarity between the claim before and after prepending a trigger, we use a large RoBERTa model fine-tuned on the Semantic Textual Similarity Task.[2] For further details, we refer the reader to §3.6.1.

3.3.2 Universal Adversarial Triggers Method

The Universal Adversarial Triggers method is developed to find n-gram trigger tokens $\mathbf{t}_{\mathbf{ff}}$, which, appended to the original input x, $f(x) = y$, cause the model to predict a target class $\widetilde{y}$: $f(t_\alpha, x) = \widetilde{y}$. In our work, we generate unigram triggers, as generating longer triggers would require additional objectives to later produce well-formed adversarial claims. We start by initializing the triggers with the token 'a'. Then, we update the embeddings of the initial trigger tokens $\mathbf{e}_\alpha$ with embeddings $\mathbf{e}_{w_i}$ of candidate adversarial trigger tokens w_i that minimize the loss $\mathcal{L}$ for the target class $\widetilde{y}$. Following the HotFlip algorithm, we reduce the brute-force optimization problem using a first-order Taylor approximation around the initial trigger embeddings:

$$\underset{w_i \in V}{arg\,min}\,[\mathbf{e}_{w_i} - \mathbf{e}_\alpha]^\top \nabla_{\mathbf{e}_\alpha} \mathcal{L} \tag{3.1}$$

where $\mathcal{V}$ is the vocabulary of the RoBERTa model and $\nabla_{\mathbf{e}_\alpha}\mathcal{L}$ is the average gradient of the task loss accumulated for all batches. This approximation allows for a $\mathcal{O}(|\mathcal{V}|)$ space complexity of the brute-force candidate trigger search.

While HotFlip finds universal adversarial triggers that successfully fool the model for many instances, we find that the most potent triggers are often negation words, e.g., 'not', 'neither', 'nowhere'. Such triggers change the meaning of the text, making the prediction of the target class correct. Ideally, adversarial triggers would preserve the original label of the claim. To this end, we propose to include an auxiliary STS model objective when searching for candidate triggers. The additional objective is used to minimize the loss $\mathcal{L}'$ for the maximum similarity score (5 out of 0) between the original claim and the claim with the prepended trigger. Thus, we arrive at the combined optimization problem:

$$\underset{w_i \in V}{arg\,min}([\mathbf{e}_{w_i} - \mathbf{e}_\alpha]^\top \nabla_{\mathbf{e}_\alpha} \mathcal{L} + [\mathbf{o}_{w_i} - \mathbf{o}_\alpha]^\top \nabla_{\mathbf{o}_\alpha} \mathcal{L}') \tag{3.2}$$

where $\mathbf{o}_w$ is the STS model embedding of word w. For the initial trigger token, we use "[MASK]" as STS selects candidates from the neighborhood of the initial token.

[2] https://huggingface.co/SparkBeyond/roberta-large-sts-b

3.3.3 Claim Generation

In addition to finding highly potent adversarial triggers, it is also of interest to generate coherent statements containing the triggers. To accomplish this, we use the HuggingFace implementation of the GPT-2 language model (Radford et al., 2019; Wolf et al., 2019), a large transformer-based language model trained on 40GB of text. The objective is to generate a coherent claim, which either entails, refutes, or is unrelated a given piece of evidence, while also including trigger words.

The language model is first fine tuned on the FEVER FC corpus with a specific input format. FEVER consists of claims and evidence with the labels `SUPPORTS`, `REFUTES`, or `NOT ENOUGH INFO` (NEI). We first concatenate evidence and claims with a special token. Next, to encourage generation of claims with certain tokens, a sequence of tokens separated by commas is prepended to the input. For training, the sequence consists of a single token randomly selected from the original claim, and four random tokens from the vocabulary. This encourages the model to only select the one token most likely to form a coherent and correct claim. The final input format is `[trigger tokens]`||`[evidence]`||`[claim]`. Adversarial claims are then generated by providing an initial input of a series of five comma-separated trigger tokens plus evidence, and progressively generating the rest of the sequence. Subsequently, the set of generated claims is pruned to include only those which contain a trigger token, and constitute the desired label. The latter is ensured by passing both evidence and claim through an external NLI model trained on SNLI (Bowman et al., 2015).

3.4 Results

We present results for universal adversarial trigger generation and coherent claim generation. Results are measured using the original FC model on claims with added triggers and generated claims (macro F_1). We also measure how well the added triggers maintain the claim's original label (semantic similarity score), the perplexity (PPL) of the claims with prepended triggers, and the semantic quality of generated claims (manual annotation). PPL is measured with a pretrained RoBERTa LM.

3.4.1 Adversarial Triggers

Table 3.1 presents the results of applying universal adversarial triggers to claims from the source class. The top-performing triggers for each direction are found in §3.6.2. The adversarial method with a single FC objective successfully deteriorates model performance by a margin of 0.264 F_1 score overall. The biggest performance decrease is when the adversarial triggers are constructed to flip the predicted class from SUPPORTS to REFUTES. We also find that 8 out of 18 triggers from the top-3 triggers

Class	F_1	STS	PPL
		No Triggers	
All	.866	5.139	11.92 (±45.92)
S	.938	5.130	12.22 (±40.34)
R	.846	5.139	12.14 (±37.70)
NEI	.817	5.147	14.29 (±84.45)
		FC Objective	
All	.602 (±.289)	4.586 (±.328)	12.96 (±55.37)
S→R	.060 (±.034)	4.270 (±.295)	12.44 (±41.74)
S→NEI	.611 (±.360)	4.502 (±.473)	12.75 (±40.50)
R→S	.749 (±.027)	4.738 (±.052)	11.91 (±36.53)
R→NEI	.715 (±.026)	4.795 (±.094)	11.77 (±36.98)
NEI→R	.685 (±.030)	4.378 (±.232)	14.20 (±83.32)
NEI→S	.793 (±.054)	4.832 (±.146)	14.72 (±93.15)
		FC+STS Objectives	
All	.763 (±.123)	4.786 (±.156)	12.97 (±58.30)
S→R	.702 (±.237)	4.629 (±.186)	12.62 (±41.91)
S→NEI	.717 (±.161)	4.722 (±.152)	12.41 (±39.66)
R→S	.778 (±.010)	4.814 (±.141)	11.93 (±37.04)
R→NEI	.779 (±.009)	4.855 (±.098)	12.20 (±37.67)
NEI→R	.780 (±.078)	4.894 (±.115)	15.27 (±111.2)
NEI→S	.821 (±.008)	4.800 (±.085)	13.42 (±82.30)

Table 3.1: Universal Adversarial Trigger method performance. Triggers are generated given claims from a source class to fool the classifier to predict a target class (column *Class*, with SUPPORTS (S), REFUTES (R), NEI). The results are averaged over the top 10 triggers.

for each direction, are negation words such as ‘nothing’, ‘nobody’, ‘neither’, ‘nowhere’ (see Table 3.4 in the appendix). The first of these triggers decreases the performance of the model to 0.014 in F_1. While this is a significant performance drop, these triggers also flip the meaning of the text. The latter is again indicated by the decrease of the semantic similarity between the claim before and after prepending a trigger token, which is the largest for the SUPPORTS to REFUTES direction. We hypothesise that the success of the best performing triggers is partly due to the meaning of the text being flipped.

Including the auxiliary STS objective increases the similarity between the claim before and after prepending the trigger for five out of six directions. Moreover, we find that now only one out of the 18 top-3 triggers for each direction are negation words. Intuitively, these adversarial triggers are worse at fooling the FC model as they also have to preserve the label of the original claim. Notably, for the SUPPORTS to REFUTES direction the trigger performance is decreased with a margin of 0.642

Target	F_1	Avg Quality	# Examples
FC Objective			
Overall	0.534	4.33	156
SUPPORTS	0.486	4.79	39
REFUTES	0.494	4.70	32
NEI	0.621	3.98	85
FC+STS Objectives			
Overall	0.635	4.63	156
SUPPORTS	0.617	4.77	67
REFUTES	0.642	4.68	28
NEI	0.647	4.44	61

Table 3.2: FC performance for generated claims.

compared to the single FC objective. We conclude that including the STS objective for generating Universal Adversarial triggers helps to preserve semantic similarity with the original claim, but also makes it harder to both find triggers preserving the label of the claim while substantially decreasing the performance of the model.

3.4.2 Generation

We use the method described in §3.3.3 to generate 156 claims using triggers found with the additional STS objective, and 156 claims without. 52 claims are generated for each class (26 flipping to one class, 26 flipping to the other). A different GPT-2 model is trained to generate claims for each specific class, with triggers specific to attacking that class used as input. The generated claims are annotated manually (see §3.6.5 for the procedure). The overall average claim quality is 4.48, indicating that most generated statements are highly semantically coherent. The macro F_1 of the generative model w.r.t. the intended label is 58.9 overall. For the model without the STS objective, the macro F_1 is 56.6, and for the model with the STS objective, it is 60.7, meaning that using triggers found with the STS objective helps the generated claims to retain their intended label.

We measure the performance of the original FC model on generated claims (Table 3.2). We compare between using triggers that are generated with the STS objective (Ex2) and without (Ex1). In both cases, the adversarial claims effectively fool the FC model, which performs 38.4% worse and 23.8% worse on Ex1 and Ex2, respectively. Additionally, the overall sentence quality increases when the triggers are found with the STS objective (Ex2). The FC model's performance is higher on claims using triggers generated with the STS objective but still significantly worse than on the original claims. We provide examples of generated claims with their evidence in Table 3.3.

Comparing FC performance with our generated claims vs. those from the development set of adversarial claims from the FEVER shared task , we see similar drops in performance (0.600 and 0.644 macro F_1, respectively). While the adversarial triggers from FEVER cause a larger performance drop, they were manually selected to meet the label coherence and grammatical correctness requirements. Conversely, we automatically generate claims that meet these requirements.

Evidence	Triggers	Generated Claim
	SUPPORTS Claims	
Since the 19th century, some Romani have also migrated to the Americas.	don, already, more, during, home	Romani have moved to the Americas during the 19th century.
Cyprus is a major tourist destination in the Mediterranean.	foreign, biggest, major, every, friends	Cyprus is a major tourist destination.
The first Nobel Prize in Chemistry was awarded in 1901 to Jacobus Henricus van't Hoff, of the Netherlands, "for his discovery of the laws of chemical dynamics and osmotic pressure in solutions."	later, already, quite, altern, whereas	Henricus Van't Hoff was already awarded the Nobel Prize.
	REFUTES Claims	
California Attorney General Kamala Harris defeated Sanchez , 61.6% to 38.4%.	phys, incarn, not, occasionally, something	Kamala Harris did not defeat Sanchez, 61.6% to 38.4%.
Uganda is in the African Great Lakes region.	unless, endorsed, picks, pref, against	Uganda is against the African Great Lakes region.
Times Higher Education World University Rankings is an annual publication of university rankings by Times Higher Education (THE) magazine.	interested, reward, visit, consumer, conclusion	Times Higher Education World University Rankings is a consumer magazine.
	NOT ENOUGH INFO Claims	
The KGB was a military service and was governed by army laws and regulations, similar to the Soviet Army or MVD Internal Troops.	nowhere, only, none, no, nothing	The KGB was only controlled by a military service.
The series revolves around Frank Castle, who uses lethal methods to fight crime as the vigilante "the Punisher", with Jon Bernthal reprising the role from Daredevil.	says, said, take, say, is	Take Me High is about Frank Castle's use of lethal techniques to fight crime.
The Suite Life of Zack & Cody is an American sitcom created by Danny Kallis and Jim Geoghan.	whilst, interest, applic, someone, nevertheless	The Suite Life of Zack & Cody was created by someone who never had the chance to work in television.

Table 3.3: Examples of generated adversarial claims. These are all claims which the FC model incorrectly classified.

3.5 Conclusion

We present a method for automatically generating highly potent, well-formed, label cohesive claims for FC. We improve upon previous work on universal adversarial triggers by determining how to construct valid claims containing a trigger word. Our method is fully automatic, whereas previous work on generating claims for fact checking is generally rule-based or requires manual intervention. As FC is only one test bed for adversarial attacks, it would be interesting to test this method on other

NLP tasks requiring semantic understanding such as question answering to better understand shortcomings of models.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

3.6 Appendices

3.6.1 Implementation Details

Models. The RoBERTa FC model (125M parameters) is fine-tuned with a batch size of 8, learning rate of 2e-5 and for a total of 4 epochs, where the epoch with the best performance is saved. We used the implementation provided by HuggingFace library. We performed a grid hyper-parameter search for the learning rate between the values 1e-5, 2e-5, and 3e-5. The average time for training a model with one set of hyperparameters is 155 minutes (± 3). The average accuracy over the different hyperparameter runs is 0.862(± 0.005) F_1 score on the validation set.

For the models that measure the perplexity and the semantical similarity we use the pretrained models provided by HuggingFace– RoBERTa large model (125M parameters) fine tuned on the STS-b task and RoBERTa base model (355M parameters) pretrained on a LM objective.

We used the HuggingFace implementation of the small GPT-2 model, which consists of 124,439,808 parameters and is fine-tuned with a batch size of 4, learning rate of 3e-5, and for a total of 20 epochs. We perform early stopping on the loss of the model on a set of validation data. The average validation loss is 0.910. The average runtime for training one of the models is 31 hours and 28 minutes.

We note that, the intermediate models used in this work and described in this section, are trained on large relatively general-purpose datasets. While, they can make some mistakes, they work well enough and using them, we don't have to rely on additional human annotations for the intermediate task.

Adversarial Triggers. The adversarial triggers are generated based on instances from the validation set. We run the algorithm for three epochs to allow for the adversarial triggers to converge. At each epoch the initial trigger is updated with the best performing trigger for the epoch (according to the loss of the FC or FC+STS objective). At the last step, we select only the top 10 triggers and remove any that have a negative loss. We choose the top 10 triggers as those are the most potent ones, adding more than top ten of the triggers preserves the same tendencies in the results, but smooths them as further down the list of adversarial attacks, the triggers do not decrease the performance of the model substantially. This is also supported by related literature (Wallace et al., 2019a), where only the top few triggers are selected.

The adversarial triggers method is run for 28.75 (± 1.47) minutes for with the FC objective and 168.6 (± 28.44) minutes for the FC+STS objective. We perform the trigger generation with a batch size of four. We additionally normalize the loss for each objective to be in the range [0,1] and also re-weight the losses with a wieht of 0.6 for the FC loss and a weight of 0.4 for the SST loss as when generated with an equal weight, the SST loss tends to preserve the same initial token in all epochs.

Datasets. The datasets used for training the FC model consist of 161,249 SUPPORTS, 60,227 REFUTES, and 69,885 NEI claims for the training split; 6,207 SUPPORTS, 6,235 REFUTES, and 6,554 NEI claims for the dev set; 6,291 SUPPORTS, 5,992 REFUTES, and 6522 NEI claims. The evidence for each claim is the gold evidence provided from the FEVER dataset, which is available for REFUTES and SUPPORTS claims. When there is more than one annotation of different evidence sentences for an instance, we include them as separate instances in the datasets. For NEI claims, we use the system of Malon (2018) to retrieve evidence sentences.

3.6.2 Top Adversarial Triggers

Table 3.4 presents the top adversarial triggers for each direction found with the Universal Adversarial Triggers method. It offers an additional way of estimating the effectiveness of the STS objective by comparing the number of negation words generated by the basic model (8) and the STS objective (2) in the top-3 triggers for each direction.

3.6.3 Computing Infrastructure

All experiments were run on a shared cluster. Requested jobs consisted of 16GB of RAM and 4 Intel Xeon Silver 4110 CPUs. We used two NVIDIA Titan RTX GPUs with 12GB of RAM for training GPT-2 and one NVIDIA Titan X GPU with 8GB of RAM for training the FC models and finding the universal adversarial triggers.

3.6.4 Evaluation Metrics

The primary evaluation metric used was macro F_1 score. We used the sklearn implementation of `precision_recall_fscore_support`, which can be found here: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics`. Briefly:

$$p = \frac{tp}{tp + fp}$$

Class	Trigger	F_1	STS	PPL
		FC Objective		
S→R	only	0.014	4.628	11.660 (36.191)
S→R	nothing	0.017	4.286	13.109 (56.882)
S→R	nobody	0.036	4.167	12.784 (37.390)
S→NEI	neither	0.047	3.901	11.509 (31.413)
S→NEI	none	0.071	4.016	13.136 (39.894)
S→NEI	Neither	0.155	3.641	11.957 (44.274)
R→S	some	0.687	4.694	11.902 (33.348)
R→S	Sometimes	0.724	4.785	10.813 (32.058)
R→S	Some	0.743	4.713	11.477 (37.243)
R→NEI	recommended	0.658	4.944	12.658 (36.658)
R→NEI	Recommend	0.686	4.789	10.854 (32.432)
R→NEI	Supported	0.710	4.739	11.972 (40.267)
NEI→R	Only	0.624	4.668	12.939 (57.666)
NEI→R	nothing	0.638	4.476	11.481 (48.781)
NEI→R	nobody	0.678	4.361	16.345 (111.60)
NEI→S	nothing	0.638	4.476	18.070 (181.85)
NEI→S	existed	0.800	4.950	15.552 (79.823)
NEI→S	area	0.808	4.834	13.857 (93.295)
		FC+STS Objectives		
S→R	never	0.048	4.267	12.745 (50.272)
S→R	every	0.637	4.612	13.714 (51.244)
S→R	didn	0.719	4.986	12.416 (41.080)
S→NEI	always	0.299	4.774	11.906 (35.686)
S→NEI	every	0.637	4.612	12.222 (38.440)
S→NEI	investors	0.696	4.920	12.920 (42.567)
R→S	over	0.761	4.741	12.139 (33.611)
R→S	about	0.765	4.826	12.052 (37.677)
R→S	her	0.774	4.513	12.624 (41.350)
R→NEI	top	0.757	4.762	12.787 (39.418)
R→NEI	also	0.770	5.034	11.751 (35.670)
R→NEI	when	0.776	4.843	12.444 (37.658)
NEI→R	only	0.562	4.677	14.372 (83.059)
NEI→R	there	0.764	4.846	11.574 (42.949)
NEI→R	just	0.786	4.916	16.879 (135.73)
NEI→S	of	0.802	4.917	11.844 (55.871)
NEI→S	is	0.815	4.931	17.507 (178.55)
NEI→S	A	0.818	4.897	12.526 (67.880)

Table 3.4: Top-3 triggers found with the Universal Adversarial Triggers methods. The triggers are generated given claims from a source class (column *Class*), so that the classifier is fooled to predict a different target class. The classes are SUPPORTS (S), REFUTES (R), NOT ENOUGH INFO (NEI).

$$r = \frac{tp}{tp + fn}$$

$$F_1 = \frac{2 * p * r}{p + r}$$

where tp are true positives, fp are false positives, and fn are false negatives.

3.6.5 Manual Evaluation

After generating the claims, two independent annotators label the overall claim quality (score of 1-5) and the true label for the claim. The inter-annotator agreement for the quality label using Krippendorff's alpha is 0.54 for the quality score and 0.38 for the claim label. Given this, we take the average of the two annotator's scores for the final quality score and have a third expert annotator examine and select the best label for each contested claim label.

Part III

Explainability for Complex Reasoning Tasks over Text

Generating Fact Checking Explanations

4

4.1 Introduction

When a potentially viral news item is rapidly or indiscriminately published by a news outlet, the responsibility of verifying the truthfulness of the item is often passed on to the audience. To alleviate this problem, independent teams of professional fact checkers manually verify the veracity and credibility of common or particularly check-worthy statements circulating the web. However, these teams have limited resources to perform manual fact checks, thus creating a need for automating the fact checking process.

The current research landscape in automated fact checking is comprised of systems that estimate the veracity of claims based on available metadata and evidence pages. Datasets like LIAR (Wang, 2017) and the multi-domain dataset MultiFC (Augenstein et al., 2019) provide real-world benchmarks for evaluation. There are also artificial datasets of a larger scale, e.g., the FEVER (Thorne et al., 2018) dataset based on Wikipedia articles. As evident from the effectiveness of state-of-the-art methods for both real-world – 0.492 macro F_1 score (Augenstein et al., 2019), and artificial data – 68.46 FEVER score (label accuracy conditioned on evidence provided for 'supported' and 'refuted' claims) (Stammbach and Neumann, 2019), the task of automating fact checking remains a significant and poignant research challenge.

A prevalent component of existing fact checking systems is a stance detection or textual entailment model that predicts whether a piece of evidence contradicts or supports a claim (Ma et al., 2018; Mohtarami et al., 2018; Xu et al., 2018). Existing research, however, rarely attempts to directly optimise the selection of relevant evidence, i.e., the self-sufficient explanation for predicting the veracity label (Thorne et al., 2018; Stammbach and Neumann, 2019). On the other hand, Alhindi et al. (2018) have reported a significant performance improvement of over 10% macro F_1 score when the system is provided with a short human explanation of the veracity label. Still, there are no attempts at automatically producing explanations, and automating the most elaborate part of the process - producing the *justification* for the veracity prediction - is an understudied problem.

In the field of NLP as a whole, both explainability and interpretability methods have gained importance recently, because most state-of-the-art models are large, neural black-box models. Interpretability, on one hand, provides an overview of the inner

Claim: The last major oil spill from a drilling accident in America happened over 40 years ago in 1969.
Ruling Comments: (...) The last major oil spill from a drilling accident in America happened over 40 years ago in 1969. (...) The largest in volume was the Santa Barbara spill of 1969 referenced by Murdock and Johnson, in which an estimated 100,000 barrels of oil spilled into the Pacific Ocean, according to the API. The Santa Barbara spill was so big it ranked seventh among the 10 largest oil spills caused by marine well blowouts in the world, the report states. Two other U.S. spills, both in 1970, rank eighth and 10th. Fourteen marine blowouts have taken place in the U.S. between 1969 and 2007. Six of them took place after 1990 and spilled a total of nearly 13,700 barrels. (...) We interviewed three scientists who said that the impact of a spill has little to do with its volume. Scientists have proven that spills far smaller than Santa Barbara's have been devastating.
Justification: While the nation's largest oil well blowout did take place in 1969, it's not factually correct to call it the "last major oil spill". First of all, two of the largest blowouts in the world took place in the U. S. the following year. More importantly, experts agree that spills far smaller in volume to the 1969 disaster have been devastating. From a scientific perspective, Johnson's decision to single out the 1969 blowout as the last "major" one makes no sense.
Ruling: Half-True

Table 4.1: Example instance from the LIAR-PLUS dataset, with oracle sentences for generating the justification highlighted.

workings of a trained model such that a user could, in principle, follow the same reasoning to come up with predictions for new instances. However, with the increasing number of neural units in published state-of-the-art models, it becomes infeasible for users to track all decisions being made by the models. Explainability, on the other hand, deals with providing local explanations about single data points that suggest the most salient areas from the input or are generated textual explanations for a particular prediction.

Saliency explanations have been studied extensively (Adebayo et al., 2018; Arras et al., 2019; Poerner et al., 2018), however, they only uncover regions with high contributions for the final prediction, while the reasoning process still remains behind the scenes. An alternative method explored in this paper is to generate textual explanations. In one of the few prior studies on this, the authors find that feeding generated explanations about multiple choice question answers to the answer predicting system improved QA performance (Rajani et al., 2019).

Inspired by this, we research how to generate explanations for veracity prediction. We frame this as a summarisation task, where, provided with elaborate fact checking reports, later referred to as *ruling comments*, the model has to generate *veracity explanations* close to the human justifications as in the example in Table 4.1. We then explore the benefits of training a joint model that learns to generate veracity explanations while also predicting the veracity of a claim.

In summary, our **contributions** are as follows:

1. We present the first study on generating veracity explanations, showing that they can successfully describe the reasons behind a veracity prediction.
2. We find that the performance of a veracity classification system can leverage information from the elaborate ruling comments, and can be further improved by training veracity prediction and veracity explanation jointly.
3. We show that optimising the joint objective of veracity prediction and veracity explanation produces explanations that achieve better coverage and overall quality and serve better at explaining the correct veracity label than explanations learned solely to mimic human justifications.

4.2 Dataset

Existing fact checking websites publish claim veracity verdicts along with ruling comments to support the verdicts. Most ruling comments span over long pages and contain redundancies, making them hard to follow. Textual explanations, by contrast, are succinct and provide the main arguments behind the decision. PolitiFact [1] provides a summary of a claim's ruling comments that summarises the whole explanation in just a few sentences.

We use the PolitiFact-based dataset LIAR-PLUS (Alhindi et al., 2018), which contains 12,836 statements with their veracity justifications. The justifications are automatically extracted from the long ruling comments, as their location is clearly indicated at the end of the ruling comments. Any sentences with words indicating the label, which Alhindi et al. (2018) select to be identical or similar to the label, are removed. We follow the same procedure to also extract the ruling comments without the summary at hand.

We remove instances that contain fewer than three sentences in the ruling comments as they indicate short veracity reports, where no summary is present. The final dataset consists of 10,146 training, 1,278 validation, and 1,255 test data points. A claim's

[1] https://www.politifact.com/

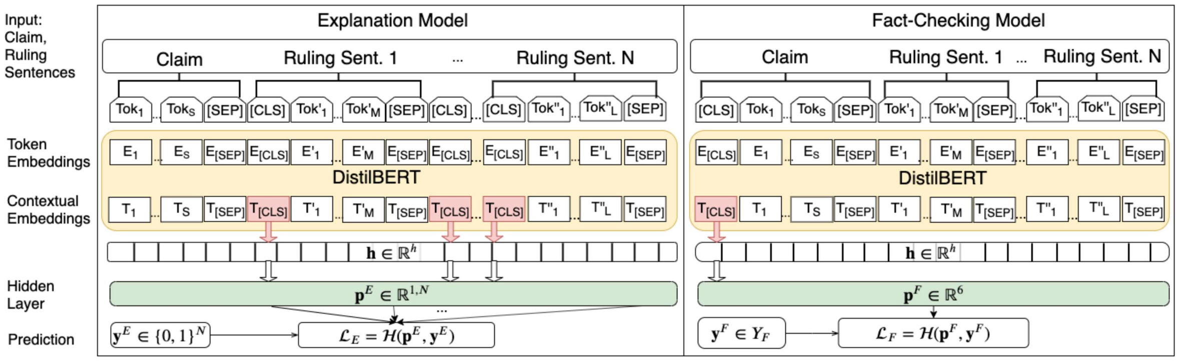

Input: Claim, Ruling Sentences
Explanation Model
Fact-Checking Model
Claim
Ruling Sent. 1
...
Ruling Sent. N
Tok1 ... TokS [SEP] [CLS] Tok'1 ... Tok'M [SEP] [CLS] ... [CLS] Tok"1 ... Tok"L [SEP]
[CLS] Tok1 ... TokS [SEP] Tok'1 ... Tok'M [SEP] ... Tok"1 ... Tok"L [SEP]
Token Embeddings
E1 ... ES E[SEP] E[CLS] E'1 ... E'M E[SEP] E[CLS] ... E[CLS] E"1 ... E"L E[SEP]
E[CLS] E1 ... ES E[SEP] E'1 ... E'M E[SEP] ... E"1 ... E"L E[SEP]
DistilBERT
Contextual Embeddings
T1 ... TS T[SEP] T[CLS] T'1 ... T'M T[SEP] T[CLS] ... T[CLS] T"1 ... T"L T[SEP]
T[CLS] T1 ... TS T[SEP] T'1 ... T'M T[SEP] ... T"1 ... T"L T[SEP]
h ∈ ℝ^h
Hidden Layer
p^E ∈ ℝ^{1,N}
p^F ∈ ℝ^6
Prediction
y^E ∈ {0, 1}^N
ℒ_E = ℋ(p^E, y^E)
y^F ∈ Y_F
ℒ_F = ℋ(p^F, y^F)

Figure 4.1: Architecture of the *Explanation* (left) and *Fact-Checking* (right) models that optimise separate objectives.

ruling comments in the dataset span over 39 sentences or 904 words on average, while the justification fits in four sentences or 89 words on average.

4.3 Method

We now describe the models we employ for training separately (1) an explanation extraction and (2) veracity prediction, as well as (3) the joint model trained to optimise both.

The models are based on DistilBERT (Sanh et al., 2019), which is a reduced version of BERT (Devlin et al., 2019) performing on par with it as reported by the authors. For each of the models described below, we take the version of DistilBERT that is pre-trained with a language-modelling objective and further fine-tune its embeddings for the specific task at hand.

4.3.1 Generating Explanations

Our explanation model, shown in Figure 4.1 (left) is inspired by the recent success of utilising the transformer model architecture for extractive summarisation (Liu and Lapata, 2019). It learns to maximize the similarity of the extracted explanation with the human justification.

We start by greedily selecting the top k sentences from each claim's ruling comments that achieve the highest ROUGE-2 F_1 score when compared to the gold justification. We choose $k = 4$, as that is the average number of sentences in veracity justifications. The selected sentences, referred to as oracles, serve as positive gold labels - $\mathbf{y}^E \in \{0,1\}^N$, where N is the total number of sentences present in the ruling comments. Appendix 4.8.1 provides an overview of the coverage that the extracted oracles achieve compared to the gold justification. Appendix 4.8.2 further presents examples of the selected oracles, compared to the gold justification.

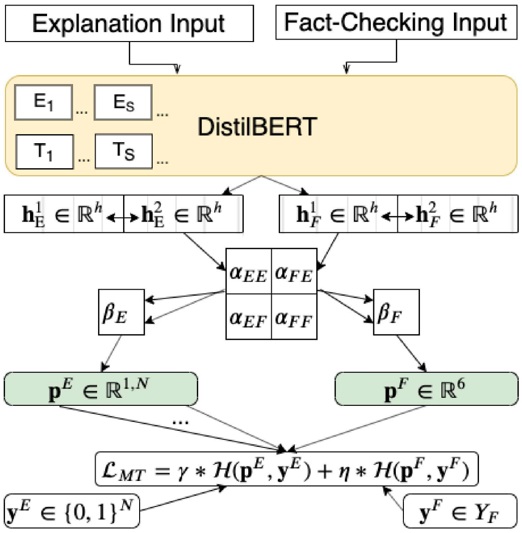

Explanation Input
Fact-Checking Input
E_1 ... E_S ...
T_1 ... T_S ...
DistilBERT
$\mathbf{h}_E^1 \in \mathbb{R}^h$
$\mathbf{h}_E^2 \in \mathbb{R}^h$
$\mathbf{h}_F^1 \in \mathbb{R}^h$
$\mathbf{h}_F^2 \in \mathbb{R}^h$
α_{EE}
α_{FE}
α_{EF}
α_{FF}
β_E
β_F
$\mathbf{p}^E \in \mathbb{R}^{1,N}$
$\mathbf{p}^F \in \mathbb{R}^6$
...
$\mathcal{L}_{MT} = \gamma * \mathcal{H}(\mathbf{p}^E, \mathbf{y}^E) + \eta * \mathcal{H}(\mathbf{p}^F, \mathbf{y}^F)$
$\mathbf{y}^E \in \{0,1\}^N$
$\mathbf{y}^F \in Y_F$

Figure 4.2: Architecture of the *Joint* model learning Explanation (E) and Fact-Checking (F) at the same time.

At training time, we learn a function $f(X) = \mathbf{p}^E$, $\mathbf{p}^E \in \mathbb{R}^{1,N}$ that, based on the input X, the text of the claim and the ruling comments, predicts which sentence should be selected - {0,1}, to constitute the explanation. At inference time, we select the top $n = 4$ sentences with the highest confidence scores.

Our extraction model, represented by function $f(X)$, takes the contextual representations produced by the last layer of DistilBERT and feeds them into a feed-forward task-specific layer - $\mathbf{h} \in \mathbb{R}^h$. It is followed by the prediction layer $\mathbf{p}^E \in \mathbb{R}^{1,N}$ with sigmoid activation. The prediction is used to optimise the cross-entropy loss function $\mathcal{L}_E = \mathcal{H}(\mathbf{p}^E, \mathbf{y}^E)$.

4.3.2 Veracity Prediction

For the veracity prediction model, shown in Figure 4.1 (right), we learn a function $g(X) = \mathbf{p}^F$ that, based on the input X, predicts the veracity of the claim $\mathbf{y}^F \in Y_F$, $Y_F =$ {*true, false, half-true, barely-true, mostly-true, pants-on-fire*}.

The function $g(X)$ takes the contextual token representations from the last layer of DistilBERT and feeds them to a task-specific feed-forward layer $\mathbf{h} \in \mathbb{R}^h$. It is followed by the prediction layer with a softmax activation $\mathbf{p}^F \in \mathbb{R}^6$. We use the prediction to optimise a cross-entropy loss function $\mathcal{L}_F = \mathcal{H}(\mathbf{p}^F, \mathbf{y}^F)$.

4.3.3 Joint Training

Finally, we learn a function $h(X) = (\mathbf{p}^E, \mathbf{p}^F)$ that, given the input X - the text of the claim and the ruling comments, predicts both the veracity explanation $\mathbf{p}^E$ and the veracity label $\mathbf{p}^F$ of a claim. The model is shown Figure 4.2. The function $h(X)$

takes the contextual embeddings $\mathbf{c}^E$ and $\mathbf{c}^F$ produced by the last layer of DistilBERT and feeds them into a cross-stitch layer (Misra et al., 2016; Ruder et al., 2019), which consists of two layers with two shared subspaces each - $\mathbf{h}_E^1$ and $\mathbf{h}_E^2$ for the explanation task and $\mathbf{h}_F^1$ and $\mathbf{h}_F^2$ for the veracity prediction task. In each of the two layers, there is one subspace for task-specific representations and one that learns cross-task representations. The subspaces and layers interact trough α values, creating the linear combinations $\widetilde{h}_E^i$ and $\widetilde{h}_F^j$, where i,j$\in \{1, 2\}$:

$$\begin{bmatrix} \widetilde{h}_E^i \\ \widetilde{h}_F^j \end{bmatrix} = \begin{bmatrix} \alpha_{EE} & \alpha_{EF} \\ \alpha_{FE} & \alpha_{FF} \end{bmatrix} \begin{bmatrix} {h_E^i}^T & {h_F^j}^T \end{bmatrix} \tag{4.1}$$

We further combine the resulting two subspaces for each task - $\widetilde{h}_E^i$ and $\widetilde{h}_F^j$ with parameters β to produce one representation per task:

$$\widetilde{h}_P^T = \begin{bmatrix} \beta_P^1 \\ \beta_P^2 \end{bmatrix}^T \begin{bmatrix} \widetilde{h}_P^1 & \widetilde{h}_P^2 \end{bmatrix}^T \tag{4.2}$$

where P $\in \{E, F\}$ is the corresponding task.

Finally, we use the produced representation to predict $\mathbf{p}^E$ and $\mathbf{p}^F$, with feed-forward layers followed by sigmoid and softmax activations accordingly. We use the prediction to optimise the joint loss function $\mathcal{L}_{MT} = \gamma * \mathcal{H}(\mathbf{p}^E, \mathbf{y}^E) + \eta * \mathcal{H}(\mathbf{p}^F, \mathbf{y}^F)$, where γ and η are used for weighted combination of the individual loss functions.

4.4 Automatic Evaluation

We first conduct an automatic evaluation of both the veracity prediction and veracity explanation models.

4.4.1 Experiments

In Table 4.3, we compare the performance of the two proposed models for generating extractive explanations. *Explain-MT* is trained jointly with a veracity prediction model, and *Explain-Extractive* is trained separately. We include the *Lead-4* system (Nallapati et al., 2017) as a baseline, which selects as a summary the first four sentences from the ruling comments. The *Oracle* system presents the best greedy approximation of the justification with sentences extracted from the ruling comments. It indicates the upper bound that could be achieved by extracting sentences from the ruling comments as an explanation. The performance of the models is measured using ROUGE-1, ROUGE-2, and ROUGE-L F_1 scores.

In Table 4.2, we again compare two models - one trained jointly - *MT-Veracity@Rul*, with the explanation generation task and one trained separately - *Veracity@Rul*. As a baseline, we report the work of Wang (2017), who train a model based on the metadata available about the claim. It is the best known model that uses only the information available from the LIAR dataset and not the gold justification, which we aim at generating.

We also provide two upper bounds serving as an indication of the approximate best performance that can be achieved given the gold justification. The first is the reported system performance from Alhindi et al. (2018), and the second - *Veracity@Just*, is our veracity prediction model but trained on gold justifications. The Alhindi et al. (2018) system is trained using a BiLSTM, while we train the *Veracity@Just* model using the same model architecture as for predicting the veracity from the ruling comments with *Veracity@Rul*.

Lastly, *Veracity@RulOracles* is the veracity model trained on the gold oracle sentences from the ruling comments. It provides a rough estimate of how much of the important information from the ruling comments is preserved in the oracles. The models are evaluated with a macro F_1 score.

4.4.2 Experimental Setup

Our models employ the base, uncased version of the pre-trained DistilBERT model. The models are fed with text depending on the task set-up - claim and ruling sentences for the explanation and joint models; claim and ruling sentences, claim and oracle sentences or claim and justification for the fact-checking model. We insert a '[CLS]' token before the start of each ruling sentence (explanation model), before the claim (fact-checking model), or at the combination of both for the joint model. The text sequence is passed through a number of Transformer layers from DistilBERT. We use the '[CLS]' embeddings from the final contextual layer of DistilBERT and feed that in task-specific feed-forward layers $\mathbf{h} \in \mathbb{R}^h$, where h is 100 for the explanation task, 150 for the veracity prediction one and 100 for each of the joint cross-stitch subspaces. Following are the task-specific prediction layers p^E.

The size of h is picked with grid-search over {50, 100, 150, 200, 300}. We also experimented with replacing the feed-forward task-specific layers with an RNN or Transformer layer or including an activation function, which did not improve task performance.

The models are trained for up to 3 epochs, and, following Liu and Lapata (2019), we evaluate the performance of the fine-tuned model on the validation set at every 50 steps, after the first epoch. We then select the model with the best ROUGE-2 F_1 score on the validation set, thus, performing a potential early stopping. The learning rate

Model	Val	Test
Wang (2017), all metadata	0.247	0.274
Veracity@RulOracles	0.308	0.300
Veracity@Rul	0.313	0.313
MT-Veracity@Rul	**0.321**	**0.323**
Alhindi et al. (2018)@Just	0.37	0.37
Veracity@Just	**0.443**	**0.443**

Table 4.2: Results (macro F_1 scores) of the veracity prediction task on all of the six classes. The models are trained using the text from the ruling oracles (@RulOracles), ruling comment (@Rul), or the gold justification (@Just).

used is 3e-5, which is chosen with a grid search over {3e-5, 4e-5, 5e-5}. We perform 175 warm-up steps (5% of the total number of steps), after also experimenting with 0, 100, and 1000 warm-up steps. Optimisation is performed with AdamW (Loshchilov and Hutter, 2017), and the learning rate is scheduled with a warm-up linear schedule (Goyal et al., 2017). The batch size during training and evaluation is 8.

The maximum input words to DistilBERT are 512, while the average length of the ruling comments is 904 words. To prevent the loss of any sentences from the ruling comments, we apply a sliding window over the input of the text and then merge the contextual representations of the separate sliding windows, mean averaging the representations in the overlap of the windows. The size of the sliding window is 300, with a stride of 60 tokens, which is the number of overlapping tokens between two successive windows. The maximum length of the encoded sequence is 1200. We find that these hyper-parameters have the best performance after experimenting with different values in a grid search.

We also include a dropout layer (with 0.1 rate for the separate and 0.15 for the joint model) after the contextual embedding provided by the transformer models and after the first linear layer as well.

The models optimise cross-entropy loss, and the joint model optimises a weighted combination of both losses. Weights are selected with a grid search - 0.9 for the task of explanation generation and 0.1 for veracity prediction. The best performance is reached with weights that bring the losses of the individual models to roughly the same scale.

4.4.3 Results and Discussion

For each claim, our proposed joint model (see §4.3.3) provides both (i) a veracity explanation and (ii) a veracity prediction. We compare our model's performance with models that learn to optimise these objectives *separately*, as no other joint models

Model	Validation			Test		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Lead-4	27.92	6.94	24.26	28.11	6.96	24.38
Oracle	43.27	22.01	38.89	43.57	22.23	39.26
Explain-Extractive	**35.64**	**13.50**	**31.44**	**35.70**	**13.51**	**31.58**
Explain-MT	35.18	12.94	30.95	35.13	12.90	30.93

Table 4.3: Results of the veracity explanation generation task. The results are ROUGE-N F_1 scores of the generated explanation w.r.t. the gold justification.

have been proposed. Table 4.2 shows the results of veracity prediction, measured in terms of macro F_1.

Judging from the performance of both *Veracity@Rul* and *MT-Veracity@Rul*, we can assume that the task is very challenging. Even given a gold explanation (Alhindi et al. (2018) and *Veracity@Just*), the macro F_1 remains below 0.5. This can be due to the small size of the dataset and/or the difficulty of the task even for human annotators. We further investigate the difficulty of the task in a human evaluation, presented in Section 4.5.

Comparing *Veracity@RulOracles* and *Veracity@Rul*, the latter achieves a slightly higher macro F_1 score, indicating that the extracted ruling oracles, while approximating the gold justification, omit information that is important for veracity prediction. Finally, when the fact checking system is learned jointly with the veracity explanation system - *MT-Veracity@Rul*, it achieves the best macro F_1 score of the three systems. The objective to extract explanations provides information about regions in the ruling comments that are close to the gold explanation, which helps the veracity prediction model to choose the correct piece of evidence.

In Table 4.3, we present an evaluation of the generated explanations, computing ROUGE F_1 score w.r.t. gold justification. Our first model, the *Explain-Extractive* system, optimises the single objective of selecting explanation sentences. It outperforms the baseline, indicating that generating veracity explanations is possible.

Explain-Extractive also outperforms the *Explain-MT* system. While we would expect that training jointly with a veracity prediction objective would improve the performance of the explanation model, as it does for the veracity prediction model, we observe the opposite. This indicates a potential mismatch between the ruling oracles and the salient regions for the fact checking model. We also find a potential indication of that in the observed performance decrease when the veracity model is trained solely on the ruling oracles compared to the one trained on all of the ruling comments. We hypothesise that, when trained jointly with the veracity extraction component, the explanation model starts to also take into account the actual knowledge needed to perform the fact check, which might not match the exact wording present in the oracles, thus decreasing the overall performance of the explanation system. We further investigate this in a manual evaluation of which of the systems - Explain-MT and Explain-Extractive, generates explanations with better qualities and with more information about the veracity label.

Finally, comparing the performance of the extractive models and the *Oracle*, we can conclude that there is still room for improvement of explanation systems when only considering extractive summarisation.

4.4.4 A Case Study

Table 4.4 presents two example explanations generated by the extractive vs. the multi-task model. In the first example, the multi-task explanation achieves higher ROUGE scores than the extractive one. The corresponding extractive summary contains information that is not important for the final veracity label, which also appears to affect the ROUGE scores of the explanation. On the other hand, the multi-task model, trained jointly with a veracity prediction component, selects sentences that are more important for the fact check, which in this case is also beneficial for the final ROUGE score of the explanation.

In the second example, the multi-task explanation has lower ROUGE scores than the extractive one. We observe that the gold justification contains some sentences that are not relevant to the fact check, and the extractive summary is fooled to select explanation sentences that are close to the gold summary. As a result, the explanation does not provide enough information about the chosen veracity label. The multi-task model, on the other hand, selects sentences that are also contributing to the prediction of the veracity labels. Thus, its explanation turns out to be more beneficial for the final fact check even though it has a lower ROUGE score compared to the gold justification.

4.5 Manual Evaluation

As the ROUGE score only accounts for word-level similarity between gold and predicted justifications, we also conduct a manual evaluation of the quality of the produced veracity explanations.

4.5.1 Experiments

Explanation Quality. We first provide a manual evaluation of the properties of three different types of explanations - gold justification, veracity explanation generated by the *Explain-MT*, and the ones generated by *Explain-Extractive*. We ask three annotators

Label: Mostly-true **Claim:** Says opponent U.S. Rep. Gwen Moore is Wisconsin’s most absent member of Congress, missing nearly 17 percent of the House votes in the second quarter of 2012.

Just: Sebring says Moore is the state’s "most absent" member of Congress and that she missed 17 percent of the House votes in the second quarter of 2012. Indeed, Moore missed nearly 17 percent of the votes in the second quarter. That is Moore’s worst quarter, but it also amounts to Sebring doing some cherry-picking of data. A longer period shows a lower percentage, but even that places Moore at the top of the state’s eight-member delegation.

Explain-Extr: Then he turns to the ad’s main point: "Gwen Moore is Wisconsin’s most absent member of Congress, missing nearly 17 percent of the House votes in the second quarter of 2012." That, again, compares to the median of 2.5 percent, according to GovTrack. Overall, the state delegation missed an average of 2.53 percent of the votes this session. But what about the other part of Sebring’s claim, that Moore is the "most absent" member of the state delegation?

Explain-MT: Then he turns to the ad’s main point: "Gwen Moore is Wisconsin’s most absent member of Congress, missing nearly 17 percent of the House votes in the second quarter of 2012." If you take out the first two quarters of 2012, Moore missed an average of 3.4 percent of the votes in the other five quarters of this session. In the second quarter, the one cited by Sebring, the missed votes easily put Moore at the top of the state’s eight-member House delegation. Looking at the full session, Moore missed 112 votes, or 6.98 percent.

Label: Half-true **Claim:** Of the more than 1.3 million temporary mortgage modifications, over half have defaulted.

Just: In the final full week of the U.S. Senate race, how did Rubio fare on his numbers about the mortgage modification program? Rubio said "over 1.3 million temporary work-outs, over half have now defaulted,” referring to a temporary mortgage modification program.

Explain-Extr: Over 1.3 million temporary work-outs, over half have now defaulted,” he said. There have been 500,000 permanent (modifications). Rubio also said that more than half of those 1.3 million had defaulted." Rubio: "The temporary modifications."

Explain-MT: Rubio also said that more than half of those 1.3 million had "defaulted." "Over 1.3 million temporary work-outs, over half have now defaulted,” he said. Of those permanent modifications, the majority survived while almost 29,000 were cancelled. The chart states that 699,924 trials had been cancelled – so note that is slightly more than half.

Table 4.4: Examples of the generated explanation of the extractive (Explain-Extr) and the multi-task model (Explain-MT) compared to the gold justification (Just).

to rank these explanations with the ranks 1, 2, 3, (first, second, and third place) according to four different criteria:

1. **Coverage.** The explanation contains important, salient information and does not miss any important points that contribute to the fact check.
2. **Non-redundancy.** The summary does not contain any information that is redundant/repeated/not relevant to the claim and the fact check.
3. **Non-contradiction.** The summary does not contain any pieces of information that are contradictory to the claim and the fact check.
4. **Overall.** Rank the explanations by their overall quality.

We also allow ties, meaning that two veracity explanations can receive the same rank if they appear the same.

For the annotation task set-up, we randomly select a small set of 40 instances from the test set and collect the three different veracity explanations for each of them. We did not provide the participants with information of the three different explanations and shuffled them randomly to prevent easily creating a position bias for the explanations. The annotators worked separately without discussing any details about the annotation task.

Explanation Informativeness. In the second manual evaluation task, we study how well the veracity explanations manage to address the information need of the user and if they sufficiently describe the veracity label. We, therefore, design the annotation task asking annotators to provide a veracity label for a claim based on a veracity explanation coming from the justification, the *Explain-MT*, or the *Explain-Extractive* system. The annotators have to provide a veracity label on two levels - binary classification - true or false, and six-class classification - true, false, half-true, barely-true, mostly-true, pants-on-fire. Each of them has to provide the label for 80 explanations, and there are two annotators per explanation.

4.5.2 Results and Discussion

Explanation Quality. Table 4.5 presents the results from the manual evaluation in the first set-up, described in Section 4.5, where annotators ranked the explanations according to four different criteria.

We compute Krippendorff's α inter-annotator agreement (IAA, Hayes and Krippendorff (2007)) as it is suited for ordinal values. The corresponding alpha values are 0.26 for *Coverage*, 0.18 for *Non-redundancy*, -0.1 for *Non-contradiction*, and 0.32 for *Overall*, where $0.67 < \alpha < 0.8$ is regarded as significant, but vary a lot for different domains.

Annotators	**Just**	**Explain-Extr**	**Explain-MT**
		Coverage	
All	**1.48**	1.89	1.68
1st	**1.50**	2.08	1.87
2nd	**1.74**	2.16	1.84
3rd	**1.21**	1.42	1.34
		Non-redundancy	
All	**1.48**	1.75	1.79
1st	**1.34**	1.84	1.76
2nd	**1.71**	1.97	2.08
3rd	**1.40**	1.42	1.53
		Non-contradiction	
All	1.45	**1.40**	1.48
1st	**1.13**	1.45	1.34
2nd	2.18	**1.63**	1.92
3rd	**1.03**	1.13	1.18
		Overall	
All	**1.58**	2.03	1.90
1st	**1.58**	2.18	1.95
2nd	**1.74**	2.13	1.92
3rd	**1.42**	1.76	1.82

Table 4.5: Mean Average Ranks (MAR) of the explanations for each of the four evaluation criteria. The explanations come from the gold justification (Just), the generated explanation (Explain-Extr), and the explanation learned jointly (Explain-MT) with the veracity prediction model. The lower MAR indicates a higher ranking, i.e., a better quality of an explanation. For each row, the best results are in bold, and the best results with automatically generated explanations are in blue.

We assume that the low IAA can be attributed to the fact that in ranking/comparison tasks for manual evaluation, the agreement between annotators might be affected by small differences in one rank position in one of the annotators as well as by the annotator bias towards ranking explanations as ties. Taking this into account, we choose to present the mean average recall for each of the annotators instead. Still, we find that their preferences are not in a perfect agreement and report only what the majority agrees upon. We also consider that the low IAA reveals that the task might be "already too difficult for humans". This insight proves to be important on its own as existing machine summarisation/question answering studies involving human evaluation do not report IAA scores (Liu and Lapata, 2019), thus, leaving essential details about the nature of the evaluation tasks ambiguous.

	Just	**Explain-Extr**	**Explain-MT**
↖ Agree-C	**0.403**	0.237	0.300
↘ Agree-NS	**0.065**	0.250	0.188
↘ Agree-NC	**0.064**	0.113	0.088
↘ Disagree	0.468	**0.400**	0.425

Table 4.6: Manual veracity labelling, given a particular explanation from the gold justification (Just), the generated explanation (Explain-Extr), and the explanation learned jointly (Explain-MT) with the veracity prediction model. Percentages of the dis/agreeing annotator predictions are shown, with agreement percentages split into: *correct* according to the gold label (Agree-C), *incorrect* (Agree-NC) or *insufficient information* (Agree-NS). The first column indicates whether higher (↖) or lower (↘) values are better. For each row, the best results are in bold, and the best results with automatically generated explanations are in blue.

We find that the gold explanation is ranked the best for all criteria except for *Non-contradiction*, where one of the annotators found that it contained more contradictory information than the automatically generated explanations, but Krippendorff's α indicates that there is no agreement between the annotations for this criterion.

Out of the two extractive explanation systems, *Explain-MT* ranks best in Coverage and Overall criteria, with 0.21 and 0.13 corresponding improvements in the ranking position. These results contradict the automatic evaluation in Section 4.4.3, where the explanation of *Explain-MT* had lower ROUGE F_1 scores. This indicates that an automatic evaluation might be insufficient in estimating the information conveyed by the particular explanation.

On the other hand, *Explain-Extr* is ranked higher than *Explain-MT* in terms of Non-redundancy and Non-contradiction, where the last criterion was disagreed upon, and the rank improvement for the first one is only marginal at 0.04.

This implies that a veracity prediction objective is not necessary to produce natural-sounding explanations (*Explain-Extr*), but that the latter is useful for generating better explanations overall and with higher coverage *Explain-MT*.

Explanation Informativeness. Table 4.6 presents the results from the second manual evaluation task, where annotators provided the veracity of a claim based on an explanation from one of the systems. We here show the results for binary labels, as annotators struggled to distinguish between 6 labels. The latter follows the same trends and are shown in Appendix 4.8.3.

The Fleiss' κ IAA for binary prediction is: *Just* – 0.269, *Explain-MT* – 0.345, *Explain-Extr* – 0.399. The highest agreement is achieved for *Explain-Extr*, which is supported by the highest proportion of agreeing annotations from Table 4.6. Surprisingly, the gold explanations from *Just* were most disagreed upon. Apart from that, looking at the agreeing annotations, gold explanations were found most sufficient in providing information about the veracity label and also were found to explain the correct label most of the time. They are followed by the explanations produced by *Explain-MT*. This supports the findings of the first manual evaluation, where the *Explain-MT* ranked better in coverage and overall quality than *Explain-Extr*.

4.6 Related Work

Generating Explanations. Generating textual explanations for model predictions is an understudied problem. The first study was Camburu et al. (2018), who generate explanations for the task of natural language inference. The authors explore three different set-ups: prediction pipelines with explanation followed by prediction, and prediction followed by explanation, and a joint multi-task learning setting. They find that first generating the explanation produces better results for the explanation task, but harms classification accuracy.

We are the first to provide a study on generating veracity explanations. We show that the generated explanations improve veracity prediction performance, and find that jointly optimising the veracity explanation and veracity prediction objectives improves the coverage and the overall quality of the explanations.

Fact Checking Interpretability. Interpreting fact checking systems has been explored in a few studies. Shu et al. (2019) study the interpretability of a system that fact checks full-length news pages by leveraging user comments from social platforms. They propose a co-attention framework, which selects both salient user comments and salient sentences from news articles. Yang et al. (2019) build an interpretable fact-checking system XFake, where shallow student and self-attention, among others, are used to highlight parts of the input. This is done solely based on the statement without considering any supporting facts. In our work, we research models that generate human-readable explanations, and directly optimise the quality of the produced explanations instead of using attention weights as a proxy. We use the LIAR dataset to train such models, which contains fact checked single-sentence claims that already contain professional justifications. As a result, we make an initial step towards automating the generation of professional fact checking justifications.

Veracity Prediction. Several studies have built fact checking systems for the LIAR dataset (Wang, 2017). The model proposed by Karimi et al. (2018) reaches 0.39 accuracy by using metadata, ruling comments, and justifications. Alhindi et al. (2018) also trains a classifier, that, based on the statement and the justification, achieves 0.37 accuracy. To the best of our knowledge, Long et al. (2017) is the only system that, without using justifications, achieves a performance above the baseline of Wang

(2017), an accuracy of 0.415—the current state-of-the-art performance on the LIAR dataset. Their model learns a veracity classifier with speaker profiles. While using metadata and external speaker profiles might provide substantial information for fact checking, they also have the potential to introduce biases towards a certain party or a speaker.

In this study, we propose a method to generate veracity explanations that would explain the reasons behind a certain veracity label independently of the speaker profile. Once trained, such methods could then be applied to other fact checking instances without human-provided explanations or even to perform end-to-end veracity prediction and veracity explanation generation given a claim.

Substantial research on fact checking methods exists for the FEVER dataset (Thorne et al., 2018), which comprises rewritten claims from Wikipedia. Systems typically perform document retrieval, evidence selection, and veracity prediction. Evidence selection is performed using keyword matching (Malon, 2018; Yoneda et al., 2018), supervised learning (Hanselowski et al., 2018; Chakrabarty et al., 2018) or sentence similarity scoring (Ma et al., 2018; Mohtarami et al., 2018; Xu et al., 2018). More recently, the multi-domain dataset MultiFC (Augenstein et al., 2019) has been proposed, which is also distributed with evidence pages. Unlike FEVER, it contains real-world claims, crawled from different fact checking portals.

While FEVER and MultiFC are larger datasets for fact checking than LIAR-PLUS, they do not contain veracity explanations and can thus not easily be used to train joint veracity prediction and explanation generation models, hence we did not use them in this study.

4.7 Conclusions

We presented the first study on generating veracity explanations, and we showed that veracity prediction can be combined with veracity explanation generation and that the multi-task set-up improves the performance of the veracity system. A manual evaluation shows that the coverage and the overall quality of the explanation system is also improved in the multi-task set-up.

For future work, an obvious next step is to investigate the possibility of generating veracity explanations from evidence pages crawled from the Web. Furthermore, other approaches of generating veracity explanations should be investigated, especially as they could improve fluency or decrease the redundancy of the generated text.

Evidence Source	ROUGE-1			ROUGE-2			ROUGE-L		
	P	R	F_1	P	R	F_1	P	R	F_1
Ruling	8.65	78.65	14.84	3.53	33.76	6.16	8.10	74.14	13.92
Ruling Oracle	43.97	49.24	43.79	22.45	24.50	22.03	39.70	44.10	39.37

Table 4.7: Comparison of sources of evidence - Ruling Comments and Ruling Oracles comapred to the target justification summary.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

4.8 Appendices

4.8.1 Comparison of different sources of evidence

Table 4.7 provides an overview of the ruling comments and the ruling oracles compared to the justification. The high recall in both ROUGE-1 and ROUGE-F achieved by the ruling comments indicates that there is a substantial coverage, i.e. over 70% of the words and long sequences in the justification can be found in the ruling comments. On the other hand, there is a small coverage for the bi-grams. Selecting the oracles from all of the ruling sentences increases ROUGE-F_1 scores mainly by improving the precision.

4.8.2 Extractive Gold Oracle Examples

Table 4.8 presents examples of selected oracles that serve as gold labels during training the extractive summarization model. The three examples represent oracles with different degrees of matching the gold summary. The first row presents an oracle that matches the gold summary with a ROUGE-L F_1 score of 60.40 compared to the gold summary. It contains all of the important information from the gold summary and even points precise, not rounded, numbers. The next example has a ROUGE-L F_1 score of 43.33, which is close to the average ROUGE-L F_1 score for the oracles. The oracle again conveys the main points from the gold justification, thus, being sufficient for the claim's explanation. Finally, the third example is of an oracle with a ROUGE-L F_1 score of 25.59. The selected oracle sentences still succeed in presenting the main points from the gold justification, which is at a more detailed level presenting specific findings. The latter might be found as a positive consequence as it presents the particular findings of the journalist that led to selecting the veracity label.

Claim: “The president promised that if he spent money on a stimulus program that unemployment would go to 5.7 percent or 6 percent. Those were his words.”
Label: Mostly-False
Just: Bramnick said “the president promised that if he spent money on a stimulus program that unemployment would go to 5.7 percent or 6 percent. Those were his words.” Two economic advisers estimated in a 2009 report that with the stimulus plan, the unemployment rate would peak near 8 percent before dropping to less than 6 percent by now. Those are critical details Bramnick’s statement ignores. To comment on this ruling, go to NJ.com.
Oracle: “The president promised that if he spent money on a stimulus program that unemployment would go to 5.7 percent or 6 percent. Those were his words,” Bramnick said in a Sept. 7 interview on NJToday. But with the stimulus plan, the report projected the nation’s jobless rate would peak near 8 percent in 2009 before falling to about 5.5 percent by now. So the estimates in the report were wrong.

Claim: The Milwaukee County bus system has “among the highest fares in the nation.”
Label: False
Just: Larson said the Milwaukee County bus system has “among the highest fares in the nation.” But the system’s’ $2.25 cash fare wasn’t at the top of a national comparison, with fares reaching as high as $4 per trip. And regular patrons who use a Smart Card are charged just $1.75 a ride, making the Milwaukee County bus system about on par with average costs.
Oracle: Larson said the Milwaukee County bus system has “among the highest fares in the nation.” Patrons who get a Smart Card pay $1.75 per ride. At the time, nine cities on that list charged more than Milwaukee’s $2.25 cash fare. The highest fare – in Nashville – was $4 per ride.

Claim: “The Republican who was just elected governor of the great state of Florida paid his campaign staffers, not with money, but with American Express gift cards.”
Label: Half-True
Just: First, we think many people might think Maddow was referring to all campaign workers, but traditional campaign staffers – the people working day in and day out on the campaign – were paid by check, like any normal job. A Republican Party official said it was simply an easier, more efficient and quicker way to pay people. And second, it’s not that unusual. In 2008, Obama did the same thing.
Oracle: “It’s a simpler and quicker way of compensating short-term help.” Neither Conston nor Burgess said how many temporary campaign workers were paid in gift cards. When asked how he was paid, Palecheck said: “Paid by check, like any normal employee there.” In fact, President Barack Obama’s campaign did the same thing in 2008.

Table 4.8: Examples of the extracted oracle summaries (Oracle) compared to the gold justification (Just).

	Just	Explain-Extr	Explain-MT
↖ Agree-C	**0.208**	0.138	0.163
↘ Agree-NS	**0.065**	0.250	0.188
↘ Agree-NC	**0.052**	0.100	0.075
↘ Disagree	0.675	**0.513**	0.575

Table 4.9: Manual classification of veracity label - true, false, half-true, barely-true, mostly-true, pants-on-fire, given a particular explanations from the gold justification (Just), the generated explanation (Explain-Extr) and the explanation learned jointly with the veracity prediction model (Explain-MT). Presented are percentages of the dis/agreeing annotator predictions, where the agreement percentages are split to: correct according to the gold label (Agree-C) , incorrect (Agree-NC) or with not sufficient information (Agree-NS). The first column indicates whether higher (↖) or lower (↘) values are better. At each row, the best set of explanations is in bold and the best automatic explanations are in blue.

4.8.3 Manual 6-way Veracity Prediction from explanations

The Fleiss' κ agreement for the 6-label manual annotations is: 0.20 on the *Just* explanations, 0.230 on the *Explain-MT* explanations, and 0.333 on the *Explain-Extr* system. Table 4.9 represent the results of the manual veracity prediction with six classes.

5 Generating Fluent Fact Checking Explanations with Unsupervised Post-Editing

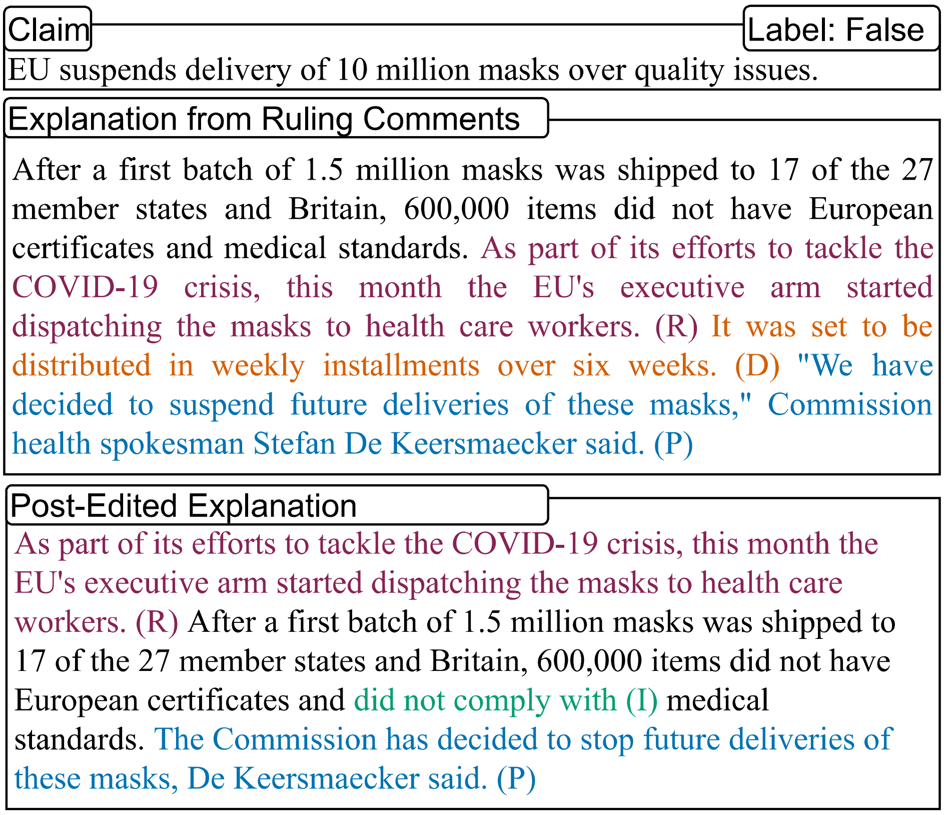
5.1 Introduction

In today's era of social media, the spread of news is a click away, regardless of whether it is fake or real. However, the quick propagation of fake news has repercussions on peoples' lives. To alleviate them, professional fact checkers manually verify the veracity and credibility of news, which is time and labor-intensive, making the process expensive and less scalable. Therefore, the need for accurate, scalable, and explainable automatic fact checking (FC) systems is inevitable (Kotonya and Toni, 2020b).

Current automatic fact checking systems perform veracity prediction for given claims based on evidence documents (Thorne et al., 2018; Augenstein et al., 2019), or based on long lists of supporting ruling comments (RCs, Wang (2017); Alhindi et al. (2018)). RCs are in-depth explanations for predicted veracity labels, but they are challenging to read and not useful as explanations for human readers due to their sizable content. Recent work (Atanasova et al., 2020c; Kotonya and Toni, 2020a) has thus proposed to use automatic summarisation to select a subset of sentences from long RCs and used them as short layman explanations. However, with a purely extractive approach (Atanasova et al., 2020c), the sentences are cherry-picked from different parts of the corresponding RCs, and as a result, explanations are often disjoint and non-fluent.

While a sequence-to-sequence model trained on parallel data can partially alleviate these problems, as Kotonya and Toni (2020a) propose, it is an expensive affair due to the large amount of data and compute required to train these models. Therefore, in this work, we focus on unsupervised post-editing of explanations extracted from RCs. Recently, researchers have leveraged unsupervised post-editing to generate paraphrases (Liu et al., 2020b) and sentence simplifications (Kumar et al., 2020). However, they use short single sentences and perform a combination of exhaustive word and phrase-level edits, which has limited applicability for longer text with multiple sentences, e.g., FC explanations, due to prohibitive convergence times.

Hence, we present a *novel iterative edit-based algorithm* performing three edit operations (insertion, deletion, reorder), all at the phrase level. Fig. 5.1 illustrates



Claim

Label: False

EU suspends delivery of 10 million masks over quality issues.

Explanation from Ruling Comments

After a first batch of 1.5 million masks was shipped to 17 of the 27 member states and Britain, 600,000 items did not have European certificates and medical standards. As part of its efforts to tackle the COVID-19 crisis, this month the EU's executive arm started dispatching the masks to health care workers. (R) It was set to be distributed in weekly installments over six weeks. (D) "We have decided to suspend future deliveries of these masks," Commission health spokesman Stefan De Keersmaecker said. (P)

Post-Edited Explanation

As part of its efforts to tackle the COVID-19 crisis, this month the EU's executive arm started dispatching the masks to health care workers. (R) After a first batch of 1.5 million masks was shipped to 17 of the 27 member states and Britain, 600,000 items did not have European certificates and did not comply with (I) medical standards. The Commission has decided to stop future deliveries of these masks, De Keersmaecker said. (P)

Figure 5.1: Example of a post-edited explanation from PubHealth that was initially extracted from RCs. We illustrate four post-editing steps: reordering (R), insertion (I), deletion (D), and paraphrasing (P).

how each post-editing step contributes to creating more concise, readable, fluent, and coherent candidate explanations. Our proposed method finds the best post-edited explanation candidate according to a scoring function, ensuring its fluency and readability, semantic preservation, and conciseness quality(§5.3.2.2). To ensure that the candidate explanations are grammatically correct, we also perform grammar checking (§5.3.2.4). As a second step, we apply paraphrasing to improve further the conciseness and human readability of the explanations (§5.3.2.5). Our approach is generic and can be applied to any other application where the objective is to generate a fluent and coherent summary.

In summary, our main **contributions** are:

- To the best of our knowledge, we are the first to explore an iterative unsupervised edit-based algorithm using only phrase-level edits that leads to feasible solutions for long text inputs.
- We show how combining an iterative algorithm with grammatical corrections, and paraphrasing-based post-processing leads to fluent and easy-to-read explanations.

- We conduct extensive experiments on the LIAR-PLUS (Wang, 2017) and PubHealth (Kotonya and Toni, 2020a) FC datasets. Our automated evaluation confirms the success of our proposed method in preserving the semantics important for the fact check and enhancing the readability of the generated explanations. Our manual evaluation confirms that our approach improves the generated explanations' fluency and conciseness.

5.2 Related Work

The most closely related work are explainable FC, generative approaches to explainability, and post-editing for language generation.

5.2.1 Explainable Fact Checking

Recent work has produced fact-checking explanations by highlighting words in tweets using neural attention (Lu and Li, 2020). Wu et al. (2020) propose to model evidence documents with decision trees, which are inherently interpretable ML models. Recently, Atanasova et al. (2020c) propose to generate free-text explanations for political claims jointly with predicting the veracity of claims. They formulate an extractive summarisation task to select a few important sentences from a long FC report. Training the summarisation task jointly with veracity prediction results in summaries that better explain the correct veracity label. Atanasova et al. (2021) also perform extractive explanation generation guided by a set of diagnostic properties of explanations and evaluate on the FEVER (Thorne et al., 2018) FC dataset, where explanation sentences are extracted from Wikipedia documents.

In the domain of public health claims, Kotonya and Toni (2020a) propose to generate explanations separately from the task of veracity prediction. Mishra et al. (2020) generate summaries of evidence documents from the Web using an attention-based mechanism. Their summaries perform better than using the original evidence documents directly. Similarly to Atanasova et al. (2020c); Kotonya and Toni (2020a), we present a generative approach for creating FC explanations. In contrast to related work, we propose an unsupervised post-editing approach to improve the fluency and readability of previously extracted FC explanations.

5.2.2 Generative Approaches to Explainability

While most work on explanation generation propose to highlight portions of the input (DeYoung et al., 2020a), some work studies generative approaches to explainability. Camburu et al. (2018) combine explanation generation and target prediction in a pipeline or a joint model for Natural Language Inference with free-text label explanations. Stammbach and Ash (2020) propose few-shot training for GPT-3 (Stepin et al., 2021) to explain a fact check from retrieved evidence snippets. GPT-3, however, is limited-access and has high computational costs. As in our work, Kotonya and Toni (2020a) first extract evidence sentences, which an abstractive summarisation model then summarises. In contrast, we are the first to perform unsupervised post-editing of explanations produced using automatic summarisation.

5.2.3 Post-Editing for Language Generation

Previous work has addressed unsupervised post-editing for multiple tasks like paraphrase generation (Liu et al., 2020b), sentence simplification (Kumar et al., 2020) or sentence summarisation (Schumann et al., 2020). However, all these tasks handle inputs shorter than the long multi-sentence extractive explanations that we have. Furthermore, they perform exhaustive edit operations at the word level and sometimes additionally at the phrase level, which increase computing complexity. Therefore, we present a novel method that performs a fixed number of edits only at the phrase level followed by grammar correction and paraphrasing.

5.3 Method

Our method is comprised of two steps. First, we select sentences from RCs that serve as extractive FC explanations (§5.3.1). We then apply unsupervised post-editing on the extractive explanations to improve their fluency and coherence (§5.3.2).

5.3.1 Selecting Sentences for Post-Editing

Supervised Selection. To produce supervised extractive explanations, we build models based on DistilBERT (Sanh et al., 2019) for LIAR-PLUS, and SciBERT (Beltagy et al., 2019) for PubHealth to allow for direct comparison with Atanasova et al. (2020c); Kotonya and Toni (2020a). We supervise explanation generation by k greedily selected sentences from a claim's RCs with the highest ROUGE-2 F_1 score w.r.t. the gold justification. We choose $k=4$ for LIAR-PLUS and $k=3$ for PubHealth, the average number of sentences in the gold justifications in the corresponding dataset. The selected sentences are positive gold labels, $\mathbf{y}^E \in \{0,1\}^N$, where N is the number of RC sentences. We also use the veracity labels $\mathbf{y}^F \in Y_F$ for supervision. Following Atanasova et al. (2020b), we learn a multi-task model $g(X)=(\mathbf{p}^E, \mathbf{p}^F)$. Given input X, comprised of a claim and the RCs, it predicts jointly the veracity explanation $\mathbf{p}^E$ and the veracity label $\mathbf{p}^F$, where $\mathbf{p}^E \in \mathbb{R}^{1,N}$ selects sentences for explanation, i.e. {0,1}, and $\mathbf{p}^F \in \mathbb{R}^m$, with $m=6$ for LIAR-PLUS, and $m=4$ for PubHealth. Finally, we optimise the joint cross-entropy loss $\mathcal{L}_{MT}=\mathcal{H}(\mathbf{p}^E, \mathbf{y}^E)+\mathcal{H}(\mathbf{p}^F, \mathbf{y}^F)$.

Unsupervised Selection. We experiment with unsupervised sentence selection to test the possibility of constructing fluent FC explanations in an entirely unsupervised way. We use Longformer (Beltagy et al., 2020), which was introduced for tasks with longer input, instead of the sliding-window approach used in Atanasova et al. (2020b), which is without cross-window attention. We train a model $h(X) = \mathbf{p}^F$ to predict the veracity of a claim. We optimise cross-entropy loss $\mathcal{L}_F = \mathcal{H}(\mathbf{p}^F, \mathbf{y}^F)$ and select k sentences $\mathbf{p}^{E'} \in \mathbb{R}^{1,N}$, {0, 1}, with the highest saliency scores. The saliency score of a sentence is the sum of the saliency scores of its tokens. The saliency of a token is the gradient of the input token w.r.t. the output (Simonyan et al., 2013). We select sentences using the raw gradients as Atanasova et al. (2020a) show that different gradient-based methods yield similar results. As the selection could be noisy (Kindermans et al., 2019), we consider these experiments as only complementary to the main supervised results.

5.3.2 Post-Editing

Our post-editing is completely unsupervised and operates on sentences obtained in §5.3.1. It is a search algorithm that evaluates the candidate sequences $\mathbf{p}^C$ for a given input sequence – $\mathbf{p}^E$ for supervised selection or $\mathbf{p}^{E'}$ for unsupervised selection. Below, we use $\mathbf{p}^E$ to denote both.

Given $\mathbf{p}^E$, we iteratively generate multiple candidates by performing phrase-level edits (§5.3.2.1). To evaluate a candidate explanation, we define a scoring function as a product of multiple scorers, also known as a product-of-experts model (Hinton, 2002). Our scoring function includes fluency and semantic preservation, and controls the length of the candidate explanation (§5.3.2.2). We repeat the process for n steps and select the last best-scoring candidate as our final output. We then use grammar correction (§5.3.2.4) and paraphrasing (§5.3.2.5) to further ensure conciseness and human readability.

5.3.2.1 Candidate sequence generation

We generate candidate sequences by phrase-level edits. We use the off-the-shelf syntactic parser from CoreNLP (Manning et al., 2014) to obtain the constituency tree of a candidate sequence $\mathbf{p}^C$. As $\mathbf{p}^C$ is long, we perform all operations at the phrase level. At each step t, our algorithm first randomly picks one operation – insertion, deletion, or reordering, and then randomly selects a phrase.

For **insertion**, our algorithm inserts a <MASK> token before the randomly selected phrase, and uses RoBERTa to evaluate the posterior probability of a candidate word (Li et al., 2020). This allows us to leverage the pre-training capabilities of RoBERTa and insert high-quality words that support the context of the overall explanation. Furthermore, inserting a <MASK> token before a phrase prevents breaking other phrases within the explanation, thus preserving their fluency.

The **deletion** operation deletes the randomly selected phrase. For the **reorder** operation we randomly select one phrase, which we call *reorder phrase*, and randomly select m phrases, which we call *anchor phrases*. We **reorder** each *anchor* with a *reorder phrase* and obtain m candidate sequences. The candidates are fed to GPT-2 to select the most fluent one with the fluency score given by Eq. 5.1.

5.3.2.2 Scoring Functions

The **fluency score** (f_{flu}) measures the language fluency of a candidate sequence. We use a pre-trained GPT-2 model (Radford et al., 2019). We use the joint likelihood of candidate $\mathbf{p}^C$:

$$f_{flu}(\mathbf{p}^C) = \prod_{i=1}^{n} P(\mathbf{p}_i^C|\mathbf{p}_1^C, ..., \mathbf{p}_{i-1}^C) \tag{5.1}$$

In §5.6.1 we evaluate the achieved fluency of the generated explanations through human evaluation. Additionally, as the fluency score measures the likelihood of the text according to GPT-2, which is trained on 40GB of Internet text, we assume that complex text that is not common or is not likely to appear on the Internet, would also have lower fluency score. Hence, we expect that improving the fluency of an explanation, would lead to more easily understood explanations. We evaluate the latter in §5.5.1 through automated readability scores.

Length score (f_{len}) This score encourages the generation of shorter sentences. We assume that reducing the length of the generated explanation is also beneficial for improving the readability of the explanation as it promotes shorter sentences, which are easier to read. The score is the inverse of the sequence length – longer candidate sentence have a lower scores. To control over-shortening, we reject explanations with fewer than 40 tokens. The number of tokens is a hyper-parameter chosen after fine-tuning on the validation split.

For **semantic preservation**, we compute similarities at both word and explanation level between our source explanation ($\mathbf{p}^E$) and candidate sequence ($\mathbf{p}^C$) at time-step t. The word-level semantic scorer evaluates the preserved amount of keyword information in the candidate sequence. Similarly to Li et al. (2020), we use RoBERTa (R) (Liu et al., 2019), a pre-trained masked language model, to compute a contextual representation of $\mathrm{word_i}$ in an explanation as $\mathrm{R}(\mathbf{p}_i^E, \mathbf{p}^E)$. Here, $\mathbf{p}^E = (\mathbf{p}_1^E \dots \mathbf{p}_m^E)$ is an input sequence of words. We then extract keywords from $\mathbf{p}^E$ using Rake (Rose et al., 2010) and compute a **keyword-level semantic similarity score**:

$$f_w(\mathbf{p}^E, \mathbf{p}^C) = \min_{k \in kw(\mathbf{p}^E)} \max_{\mathbf{p}_i^C \in \mathbf{p}^C} R(k, \mathbf{p}^E)^\intercal R(\mathbf{p}_i^C, \mathbf{p}^C) \tag{5.2}$$

which is the lowest cosine similarity among all keywords i.e. the least matched keyword of $\mathbf{p}^E$.

The keyword-level semantic similarity preserves the semantic information of the separate keywords in the text. It is, thus, not affected by changes in words that do not bear significant meaning for the overall explanation. However, as this semantic similarity is performed at keyword-level it does not account for preserving the overall meaning of the text and the context that the keywords are used in.

Hence, we also employ an **explanation-level semantic preservation scorer**. It measures the cosine similarity of two explanation vectors, which are explanation encodings that contain the overall semantic meaning of the explanation:

$$f_e(\mathbf{p}^E, \mathbf{p}^C) = {}^{(\mathbf{p}^C)^\intercal \mathbf{p}^E}/_{||\mathbf{p}^C|| |\mathbf{p}^E||} \tag{5.3}$$

We use SBERT (Reimers and Gurevych, 2019) for obtaining embeddings for both $\mathbf{p}^E$, $\mathbf{p}^C$. Our overall semantic score is the product of the word level and the explanation level semantics scores:

$$f_{sem}(\mathbf{p}^E, \mathbf{p}^C) = f_w(\mathbf{p}^E, \mathbf{p}^C)^\beta . f_e(\mathbf{p}^E, \mathbf{p}^C)^\eta \tag{5.4}$$

where β, and η are hyper-parameter weights for the separate scores. We evaluate the semantic preservation of the post-edited explanations with automated ROUGE scores (§5.5.2) and manual human annotations (§5.6.1, §5.6.2).

Lastly, **Named Entity (NE) score (f_{ent})** is an additional measure for meaning preservation, since NEs hold the key information within a sentence. We identify NEs using an off-the-shelf entity tagger (Honnibal and Montani, 2017) and count their number in a given explanation.

Our **overall scoring** function is the product of individual scores, where α, γ, and δ are hyper-parameter weights for the different scores:

$$f(\mathbf{p}^C) = f_{flu}(\mathbf{p}^C)^\alpha . f_{sem}(\mathbf{p}^E, \mathbf{p}^C) . f_{len}(\mathbf{p}^C)^\gamma . f_{ent}(\mathbf{p}^C)^\delta \tag{5.5}$$

5.3.2.3 Iterative Edit-based Algorithm

Given input explanations, our algorithm iteratively performs edit operations for n steps to search for a highly scored candidate ($\mathbf{p}^C$). At each step, it computes scores for the previous ($\mathbf{p}^{C-1}$) and candidate sequence (Eq. 5.5). It selects $\mathbf{p}^C$ if its score is larger than $\mathbf{p}^{C-1}$ by a multiplicative factor r_{op}:

$$f_{\mathbf{p}^C}/f_{\mathbf{p}^{C-1}} > r_{op} \tag{5.6}$$

For each edit operation, we use a separate threshold value r_{op}. r_{op} allows controlling specific operations where $r_{op} < 1$ allows the selection of candidates ($\mathbf{p}^C$) which have lower scores than $\mathbf{p}^{C-1}$. We tune all hyper-parameters, including r_{op}, n, etc., using the validation split of the LIAR-PLUS dataset.

5.3.2.4 Grammatical Correction

Once the best candidate explanation is selected, we feed it to the LanguageTool (2022) toolkit, which detects grammatical errors like capitalization and irrelevant punctuation, and returns a corrected version of the explanation. Furthermore, to ensure that we have no incomplete sentences, we remove sentences without verbs in the explanation. These two steps further ensure that the generated explanations are fluent (further evaluated in §5.6.1).

5.3.2.5 Paraphrasing

Finally, to improve fluency and readability further, we use Pegasus (Zhang et al., 2020a), a model pre-trained for abstractive text summarisation. It focuses on relevant input parts to summarise the input semantics in a concise and readable way. Since we want ourboth fluent and human-readable explanations, we leverage Pegasus without fine-tuning on downstream tasks. This way, after applying our iterative edit-based algorithm with grammatical error correction and paraphrasing, we obtain fluent, coherent, and human-readable explanations.

5.4 Experiments

5.4.1 Datasets

We use two FC datasets, LIAR-PLUS (Wang, 2017) and PubHealth (Kotonya and Toni, 2020a). These are the only real-world FC datasets that provide short veracity justifications along with claims, RCs, and veracity labels. We provide the size of the splits in Tab. 5.5, app. The LIAR-PLUS labels are {true, false, half-true, barely-true, mostly-true, pants-on-fire}, and in PubHealth, {true, false, mixture, unproven}. PubHealth is manually curated, e.g., to exclude poorly defined claims. Finally, the claims in PubHealth are more challenging to read than those in LIAR-PLUS and other real-world FC datasets.

5.4.2 Models

Our experiments include the following models; their hyper-parameters are given in Appendix 5.8.2.

(Un)Supervised Top$^{\mathrm{N}}$ extracts RC sentences in an (un)supervised way (§5.3.1), which are later used as input to our method.

(Un)Supervised Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$ generates explanations with the iterative editing (§5.3.2.3) and grammar correction (§5.3.2.4). The inputs are sentences extracted with (Un)Supervised $\mathrm{Top}^{\mathrm{N}}$.

(Un)Supervised Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$+Para generates explanations by paraphrasing the explanations from (Un)Supervised $\mathrm{Top}^{\mathrm{N}}$+$\mathrm{Edits}^{\mathrm{N}}$ (§5.3.2.5).

MTSum – Atanasova et al. (2020b), is a reference model that trains a multi-task system to predict veracity labels and extract explanation N sentences, where N is the average number of the sentences in the justifications of each dataset.

AbstrSum – Kotonya and Toni (2020a), is a baseline model that generates abstractive explanations with an average sentence length of 3.

Lead$^{\mathrm{K}}$ (Nallapati et al., 2017) is a common lower-bound baseline for summarisation models. It selects the first K sentences of the RCs.

5.4.3 Evaluation Overview

We perform both automatic and manual evaluations of the models above. We include automatic measures for readability (§5.5.1). While the latter was not included in prior work, we consider readability an essential quality of an explanation, and thus report it. We further include automatic ROUGE F_1 scores (overlap of the generated explanations with the gold ones, §5.5.2) for compatibility with prior work and to ensure that our generated explanations don't shift much from the gold ones. In particular, we are interested whether the ROUGE scores for the post-edited explanations are not significantly different from the ROUGE scores of the non-edited explanations, which would indicate preservation of the content important for FC. We note, however, that the automatic measures are limited as they are based on word-level statistics. Especially ROUGE scores should be taken with a grain of salt, as only exact word matches are scored higher and paraphrases or synonyms of words in the gold summary are not scored. Hence, we conduct a manual evaluation to further assess the quality of the generated explanations with a user study. As manual evaluation is expensive to obtain, the latter is, however, usually estimated based on small samples.

5.5 Automatic Evaluation

We use ROUGE F_1 scores to compute overlap between the generated explanations and the gold ones, and compute readability scores to assess how challenging the produced explanations are to read.

5.5.1 Readability Results

Metrics. Readability is desirable for FC explanations, as a challenging to read explanation would fail to convey the reasons for the chosen veracity label and would

	Method	Flesch↑	DC↓
	LIAR-PLUS		
Base	Lead^4	51.70	8.73
	Lead^6	53.24	8.43
Sup.	Top^6 (Sup.)	58.82	7.88
	Top^6+Edits^6	60.21	7.75
	Top^6+Edits^6+Par	66.34	7.42
Uns.	Top^6 (Uns.)	53.33	8.50
	Top^6+Edits^6	55.25	8.46
	Top^6+Edits^6+Par	62.13	8.11
	MTSum^4 (2020c)	58.56	7.99
	Justification	58.81	8.23
	PubHealth		
Base	Lead^3	44.44	9.12
	Lead^5	45.96	8.85
Sup.	Top^5 (Sup.)	48.63	8.67
	Top^5+Edits^5	53.79	8.37
	Top^5+Edits^5+Par	61.39	7.97
Uns.	Top^5 (Uns.)	45.20	8.94
	Top^5+Edits^5	50.74	8.63
	Top^5+Edits^5+Par	60.07	8.15
	MTSum^3 (2020c)	48.73	8.88
	Justification	49.29	9.17

Table 5.1: Readability measures – Flesch and Dale-Chall (DC) (§5.5.1), over the test splits. We report baseline (Base), supervised (Sup.), and unsupervised (Uns.) results (§5.3.1). We also report prior work results – MTSum^N, where we have the outputs to compute readability, and results given the gold Justification. Readability scores for Top^N+Edits^N and Top^N+Edits^N+Par are statistically significant (p<0.05) compared to Top^N and MTSum^N, except for the score in purple.

not improve the trust of end-users. We evaluate readability with Flesch Reading Ease (Kincaid et al., 1975) and Dale-Chall Readability Score (Powers et al., 1958). Flesch Reading Ease gives a text a score $\in [1, 100]$, where a score $\in [30, 50]$ requires college education and is difficult to read, a score $\in (50, 60]$ requires a 10-12th school grade and is fairly difficult to read, a score $\in (60, 70]$ is regarded as plain English, easily understood by 13-15-year-old students. Dale-Chall Readability Score uses a curated list of words familiar to lower-grade students to assess the complexity of a text. It gives a text a score $\in [9.0, 9.9]$ when it is easily understood by a 13-15th-grade (college) student, a score $\in [8.0, 8.9]$ when it is easily understood by an 11-12th-grade student, a score $\in [7.0, 7.9]$ when it is easily understood by a 9-10th-grade student. We take the mean of the readability scores for the separate instances in the test split (see validation in Tab. 5.11, app.). For 95% confidence intervals for the scores, see Tab. 5.9, app.

Results. Table 5.1 presents the readability results, where our iterative edit-based algorithm consistently improves the reading ease of the explanations by up to 5.16 points, and reduces the grade requirement by up to 0.30 points. The improvements are statistically significant (p<0.05) in both supervised and unsupervised explanations, except for the Dale-Chall score for the LIAR unsupervised explanations. Paraphrasing further improves significantly (p<0.05) the text's reading ease by up to 9.33 points, and reduces the grade requirement by up to 0.48 points. Importantly, MTSum (Atanasova et al., 2020b) explantions and the gold justifications are fairly difficult to read and can require even college education to grasp the explanation, while the explanations generated by our algorithm can be easily understood by 13-15-year-old students according to the Flesch Reading Ease score.

Overall observations. Our results show that our method makes FC explanations less challenging to read and makes them accessible to a broader audience of up to 10th-grade students.

5.5.2 Automatic ROUGE Scores

Metrics. To evaluate the generated explanations w.r.t. the gold justifications, we follow Atanasova et al. (2020b); Kotonya and Toni (2020a) and use measures from automatic text summarisation – ROUGE-1/2/L scores. These account for n-gram (1/2) and longest (L) overlap between generated and gold justification. The scores are recall-oriented, i.e., they calculate how many of the n-grams in the gold text appear in the generated one.

Caveats. Here, we use ROUGE scores to verify that the generated explanations *preserve information important for the fact check*, as opposed to generating completely unrelated text. Thus, we are interested in whether the ROUGE scores of the post-edited explanations are *close but not necessarily higher* than those of the input sentences selected from RCs. Notably, we include paraphrasing and new word insertion to improve the explanation's readability, which, while bearing the same meaning, necessarily results in lower ROUGE scores.

Results. Table 5.2 presents the ROUGE score results. First, comparing the results for the input Top^{N} sentences with the intermediate and final explanations generated by our system, we see that, while very close, the ROUGE scores tend to decrease. For PubHealth, we also see that the intermediate explanations always have higher ROUGE scores than our system's final explanations. These observations corroborate two main assumptions about our system. First, our system preserves a large portion of the information important for explaining the veracity label, which is also present in the justification. This is further corroborated by observing that the decrease in the ROUGE scores is often not statistically significant ($p < 0.05$, except for some ROUGE-2 and one ROUGE-L score). Second, the iterative editing and the subsequent paraphrasing allow for the introduction of novel n-grams, which preserve the meaning of the text, but are not explicitly present in the gold justification, which affects the word-level ROUGE scores. We further discuss this in §5.7 and the appendix.

	Method	R-1↑	R-2↑	R-L↑
	LIAR-PLUS			
Base	Lead^4	*28.11*	*6.96*	*24.38*
	Lead^6	29.15	8.28	25.84
Sup.	Top^6 (Sup.)	34.42	12.36	30.58
	Top^6+Edits^6	33.92	11.73	30.01
	Top^6+Edits^6+Par	33.94	11.25	30.08
Uns.	Top^6 (Uns.)	29.63	7.58	25.86
	Top^6+Edits^6	28.93	7.06	25.14
	Top^6+Edits^6+Par	28.98	6.84	25.39
	MTSum^4 (2020c)	*35.70*	*13.51*	*31.58*
	PubHealth			
Base	Lead^3	*29.01*	*10.24*	*24.18*
	Lead^3	23.05	6.28	19.27
	Lead^5	23.73	6.86	20.67
Sup.	Top^5 (Sup.)	29.93	12.42	26.24
	Top^5+Edits^5	29.38	11.16	25.41
	Top^5+Edits^5+Par	28.40	9.56	24.37
Uns.	Top^5 (Uns.)	23.52	6.12	19.93
	Top^5+Edits^5	23.09	5.56	19.44
	Top^5+Edits^5+Par	23.35	5.38	19.56
	AbstrSum^3 (2020a)	*32.30*	*13.46*	*26.99*
	MTSum^3 (2020c)	33.55	13.12	29.41

Table 5.2: ROUGE-1/2/L F_1 scores (§5.5.2) of baseline (Base), supervised (Sup.) and usupervised (Uns.) methods over the test splits. In *italics*, are results reported in prior work. Underlined scores of Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$ and Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$+Par are statistically significant ($p < 0.05$) compared to Top$^{\mathrm{N}}$ scores, N={5,6}. For validation and ablations (Tab. 5.12), and for confidence intervals (Tab. 5.10), see appendix.

The ROUGE scores of the explanations generated by our post-editing algorithm when fed with sentences selected in an unsupervised way are considerably lower than with the supervised models. Hence, supervision for extracting the most important sentences is important to obtain explanations close to the gold ones. Finally, the systems' results are mostly above Lead$^{\mathrm{N}}$, with a few exceptions for the unsupervised explanations for LIAR-PLUS.

Overall observations. We note that while automatic measures can serve as sanity checks and point to major discrepancies between generated explanations and gold ones, related work in generating FC explanations (Atanasova et al., 2020b) has shown that the automatic scores to some extent disagree with human evaluation studies, as they only capture word-level overlap and cannot reflect improvements of explanation quality. Human evaluations are therefore conducted for most summarisation models (Chen and Bansal, 2018; Tan et al., 2017), which we include in §5.6.

5.6 Manual Evaluation

As automated ROUGE scores only account for word-level similarity between the generated and the gold explanation, and the readability scores account only for surface-level characteristics of the explanation, we further conduct a manual evaluation of the quality of the produced explanations.

5.6.1 Explanation Quality

We manually evaluate two explanations: our baseline method (the input $\mathrm{Top}^{\mathrm{N}}$ sentences) and our best approach (the final explanations produced after paraphrasing ($\mathrm{Top}^{\mathrm{N}}$+$\mathrm{Edits}^{\mathrm{N}}$+Par)). We perform a manual evaluation of the test explanations obtained from supervised selection for both datasets with two annotators for each. Both annotators have a university-level education in English.

Metrics. We show a claim, veracity label, and two explanations to each annotator and ask them to rank the explanations according to the following criteria. **Coverage** means the explanation contains important and salient information for the fact check. **Non-redundancy** implies the explanation does not contain any redundant/repeated/not relevant information to the claim. **Non-contradiction** checks if there is information contradictory to the fact check. **Fluency** measures the grammatical correctness of the explanation and if there is a coherent story. **Overall** measures the overall explanation quality. We allow annotators to give the same rank to both explanations (Atanasova et al., 2020b). We randomly sample 40 instances[1] and do not provide the annotators the explanation type.

Results. Table 5.3 presents the human evaluation results for the first task. Each row indicates the annotator number and the number of times they ranked an explanation higher for one criterion. Our system's explanations achieve higher acceptance for non-redundancy and fluency for LIAR-PLUS. The results are more pronounced for the PubHealth dataset, where our system's explanations were preferred in almost all metrics by both annotators. We hypothesise that PubHealth being a manually curated dataset leads to overall cleaner post-editing explanations, which annotators prefer.

5.6.2 Explanation Informativeness

Metrics. We also perform a manual evaluation for veracity prediction. We ask annotators to provide a veracity label for a claim and an explanation where, as for Explanation Quality, the explanations are either our system's input or output. The annotators provide a veracity label for three-way classification: true, false, and insuffi-

[1] Due to the increased cost and execution time of the complex annotation task, and following related work that manually evaluates FC explanations (Atanasova et al., 2020b) and machine-generated summaries (Liu and Lapata, 2019).

Criterion		LIAR-PLUS			PubHealth		
	#	T^6	T^6+E^6+P	Both	T^5	T^5+E^5+P	Both
Coverage	1	**42.5**	0.0	57.5	27.5	**60.0**	12.5
	2	**40.0**	5.0	55.0	**22.5**	20.0	57.5
Non-redundancy	1	10.0	**87.5**	2.5	10.0	**82.5**	7.5
	2	7.5	**10.0**	82.5	7.5	**75.0**	17.5
Non-contradictory	1	**32.5**	5.0	62.5	7.5	**10.0**	82.5
	2	**10.0**	7.5	82.5	**20.0**	15.0	65.0
Fluency	1	40.0	**57.5**	2.5	35.0	**52.5**	12.5
	2	**77.5**	15.0	7.5	20.0	**72.5**	7.5
Fluency	1	**57.5**	42.5	0.0	35.0	**62.5**	2.5
	2	**62.5**	15.0	22.5	25.0	**67.5**	7.5

Table 5.3: Manual annotation results of explanation quality. Each value is the proportion of the times an annotator preferred a justification for a criterion. The preferred method, out of the input $\mathrm{Top}^{\mathrm{N}}$ (Supervised) and the output of our method, $\mathrm{Top}^{\mathrm{N}}$+$\mathrm{Edits}^{\mathrm{N}}$+Par, is emboldened, Both indicates no preference.

cient (see map to original labels in app.). We use 30 instances of each explanation type and perform evaluation with two annotators for each dataset and instance.

Results. For LIAR-PLUS, one annotator gave the correct label 80% times for the input and 67% times for the output explanations. The second annotator chose the correct label 56% & 44% times correspondingly (Tab. 5.4 in app.). For PubHealth, both annotators found each explanation useful for the task. The first annotator chose the correct label 50% & 40% of the times for the input and output explanations. The second annotator chose the correct label for 70% of both explanations. This corroborates that for a clean dataset like PubHealth our explanations help for the task of veracity prediction.

5.7 Discussion and Conclusion

Our automatic and manual evaluation results suggest two main implications of our post-editing algorithm. First, the automatic ROUGE evaluation confirmed that the post-editing preserves a large portion of important information contained in the gold explanation and important for FC. Our manual veracity predictions further supports this – the post-edited explanations are most useful for predicting the correct label (see also Tab. 5.13, app. for examples). Hence, we conjecture that our post-editing can be applied more generally for automated summarisation for knowledge-intensive tasks, such as FC and question answering, where the information needed for prediction has to be preserved.

#	Explanation Type	LIAR-PLUS			PubHealth		
		M	NM	I	M	NM	I
1	Top^N (Supervised)	20	5	5	15	15	0
1	Top^6+Edits^6+Par	14	7	9	12	18	0
2	Top^N (Supervised)	11	14	5	21	9	0
2	Top^5+Edits^5+Par	13	10	7	21	9	0

Table 5.4: Manual evaluation results for predicting a veracity label. # refers to annotator number, M/NM refers to number of matches/non-matches between annotator and original labels, I refers to number of explanations that were found to be insufficient to predict a label.

Second, with both the automatic and manual evaluation, we corroborate that our proposed post-editing method improves several qualities of the generated explanations – fluency, conciseness, and readability. The latter supports the usefulness of the length and fluency scores as well as the grammatical correction and the paraphrasing steps promoting these particular qualities of the generated explanations. Fluency, conciseness, and readability are important prerequisites for building trust in automated FC predictions especially for systems used in practice as Thagard (1989) find that people generally prefer simpler, more general explanations with fewer causes. They can also contribute to reaching a broader audience when conveying the claim's veracity. Conciseness and readability are also the downsides of current professional long and in-depth RCs, which some leading FC organisations, e.g., PolitiFact, have slowly started addressing by including short overview sections.

5.8 Appendices

5.8.1 Manual Evaluation

As explained in the Section 5.6 of the main paper, we mapped user inputs (TRUE / FALSE) for task two to the original labels for each dataset. For Liar, we map "true", "mostly-true", "half-true" to TRUE and "false", "pants-fire", and "barely-true" to FALSE. In the PubHealth dataset, we map "true" to TRUE, "false" to FALSE. The "insufficient" label is mapped to UNPROVEN. This way, once the mapping is done, we then compute the number of matches and non-matches to get an overall accuracy for this subset.

We appointed annotators with a university-level education in English.

Additional human evaluation results are presented in Table 5.4.

5.8.2 Experimental Setup

Table 5.5 presents split size information for the used datasets.

Dataset	Train size	Dev size	Test size
LIAR-PLUS	10,146	1,278	1,255
PubHealth	9,817	1,227	1,235

Table 5.5: Size of the fact checking datasets used in this work (§5.4.1).

	Validation			Test		
Method	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
SciBERT, w-1, l-1200	26.00	7.29	21.41	25.78	7.71	21.42
SciBERT, w-1, l-1500	27.78	9.81	23.32	27.37	9.62	23.07
SciBERT, w-1, l-1700	28.73	11.27	24.42	28.45	11.32	24.21
SciBERT, w-2, l-1700	30.15	12.32	25.66	29.71	12.04	25.35
SciBERT, w-5, l-1700	30.96	12.59	26.54	30.79	12.31	26.38

Table 5.6: Fine-tuning for PubHealth supervised multi-task model over positive sentence loss weight, base model and maximum length.

5.8.2.1 Selection of Ruling Comments

For the supervised selection of RCs, as described in Section 5.3.1, we follow the implementation of the multi-task model of Atanasova et al. (2020b). For LIAR-PLUS, we don't conduct fine-tuning as the model is already optimised for the dataset. For PubHealth, we change the base model to SciBERT, as the claims in PubHealth are from the health domain and previous work (Kotonya and Toni, 2020a) SciBERT outperforms BERTs for the domain. Table 5.6 presents the results for the fine-tuning we performed over the multi-task architecture with a grid-search over the maximum length limit of the text and the weight for the positive sentences in the explanation extraction training objective. We finally select and use explanations generated with the multi-task model with a maximum text length of 1700, and a positive sentence weight of 5.

For the unsupervised selection of explanation sentences, we employ a Longformer model. We construct the Longformer model with BERT as a base architecture and conduct 2000 additional fine-tuning steps for the newly added cross-attention weights to be optimised. We then train models for both datasets supervised by veracity prediction. The most salient sentences are selected as the sentences that have the highest sum of token saliencies.

Finally, we remove long sentences and questions from the RCs, where the ROUGE score changes after filtering are illustrated in Table 5.7, which results in the $\mathrm{Top}^{\mathrm{N}}$ sentences, that are used as input for the post-editing method.

These experiments were run on a single NVIDIA TitanRTX GPU with 24GB memory and 4 Intel Xeon Silver 4110 CPUs. Model training took ~ 3 hours.

	Validation			Test		
Method	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
LIAR-PLUS Unsup						
Top^6	29.26	7.98	25.83	29.62	7.94	26.04
Filtered Top^6	29.52	7.90	25.98	29.60	7.96	25.94
LIAR-PLUS SUP						
Top^6	34.42	12.35	30.64	34.49	12.54	30.67
Filtered Top^6	34.30	12.20	30.51	34.42	12.36	30.58
PubHealth Unsup						
Top^5	23.78	6.23	19.95	23.13	6.08	19.63
Filtered Top^5	23.94	6.13	20.04	23.52	6.12	19.93
PubHealth SUP						
Top^5	30.24	12.61	26.36	29.78	12.50	26.18
Filtered Top^5	30.35	12.63	26.43	29.93	12.42	26.24

Table 5.7: Sentence clean-up of long sentences for LIAR-PLUS and PubHealth.

5.8.2.2 Iterative Based Algorithm

We used the validation split of LIAR-PLUS to select the best hyper-parameters for both datasets. We use the weight of 1.5, 1.2, 1.4, 0.95 for α, η, γ, δ and 1.0 for β in our scoring function. We set the thresholds as 0.94 for reordering, 0.97 for deletion, and 1.10 for insertion. We keep all models – GPT-2, RoBERTa, and Pegasus, fixed and do not finetune them on any in-house dataset. We run our search algorithm on a single V100-32 GB GPU for 220 steps, which takes around 13 hours for each split for both datasets.

5.8.3 Novelty and Copy Rate

Table 5.8 presents additional statistics for the generated explanations from the test sets of both datasets. First, we compute how many of the words from the input Top$^{\text{N}}$ RCss are preserved in the final explanation. We find that with the final step of the post-editing process, up to 8% of the tokens from the RCs are not found in the final explanation. On the other hand, our post-editing approach generates up to 10% novel words that are not previously found in the RCs. This could explain the lower results for the ROUGE scores, which account only for exact token overlaps. Finally, while ROUGE scores are recall-oriented, i.e., they compute how many of the words in the gold explanation can be found in the candidate one, we compute a precision-oriented statistic of the words in the candidate that can be found in the gold explanation. Surprisingly, while ROUGE scores of our generated explanations decrease

Method	Copy Rate	Novelty	Gold Coverage
LIAR-PLUS			
Top^6 Sup.	100	0	29.2 ±11.4
Justification	41.4 ±13.0	58.6 ±13.0	100
Top^6+Edits^6 Sup.	98.5 ±1.8	1.5 ±1.8	30.7 ±12.1
Top^6+Edits^6+Par Sup.	90.8 ±4.8	9.2 ±4.8	32.5 ±12.6
PubHealth			
Top^5 Sup.	100	0	26.3 ±21.2
Justification	47.1 ±21.0	52.9 ±21.0	100
Top^5+Edits^5 Sup.	98.1 ±3.4	1.8 ±2.0	27.8 ±21.3
Top^5+Edits^5+Par Sup.	90.4 ±5.8	9.5 ±5.2	28.5 ±20.2

Table 5.8: Copy rate from the Ruling Comments, Novelty w.r.t. the Ruling comments, and Coverage % of words in the explanation that are found in the justification.

after post-processing, the reverse score increases, pointing to improvements in the precision-oriented overlap with our method.

In addition, in LIAR/PubHealth, the average summary length is 136/142 tokens for the extracted RCs, 89/86 for the gold justifications, 118.7/117.3 after iterative editing, and 98.5/94.7 after paraphrasing.

5.8.4 Automatic Evaluation

In Table 5.9 and Table 5.10, we provide more detailed results over the test splits including confidence intervals. In Table 5.11 and Table 5.12, we provide results over the validation split of the datasets for the ROUGE and readability automatic evaluation. We additionally provide ablation results for components of our approach. First, applying Pegasus directly on the extracted sentences preserves a slightly larger amount of information when compared to applying Pegasus on top of the iterative editing approach – up to 0.96 ROUGE-L scores, but the readability scores are still lower – up to 4.28 Flesch Reading Ease points. We also show results of the two parts included in the Edits step – the iterative editing and the grammar correction. We find that the grammar correction improves the ROUGE scores with up to 8 ROUGE-L score points and up to 8 Flesch Reading Ease points.

5.8.5 Case Studies

Table 5.13 presents a case study from the PubHealth dataset. Overall, the initial extracted RC sentences are transformed to be more concise, fluent and human-readable by applying the iterative post-editing algorithm followed by paraphrasing. We can also see that compared to the original explanation, the post-edited explanations contain words that do not change the semantics of the explanation, but would not be scored as

	Method	Flesch ↗	Flesch CI	Dale-Chall ↘	Dale-Chall CI
	LIAR-PLUS				
Baselines	Lead^{4}	51.70	[50.93-52.53]	8.73	[8.67-8.78]
	Lead^{6}	53.24	[52.58-53.92]	8.43	[8.38-8.47]
Supervised	Top^{6} (Supervised)	58.82	[58.13-59.54]	7.88	[8.17-8.28]
	Top^{6}+Edits^{6}	60.21	[59.51-60.95]	7.75	[7.70-7.80]
	Top^{6}+Edits^{6}+Par	66.34	[65.73-66.98]	7.42	[7.37-7.47]
Unsupervised	Top^{6} (Unsupervised)	53.33	[52.70-53.92]	8.50	[8.46-8.54]
	Top^{6}+Edits^{6}	55.25	[54.60-55.88]	8.46	[8.42-8.51]
	Top^{6}+Edits^{6}+Par	62.13	[61.56-62.71]	8.11	[8.06-8.15]
	MTSum^{4} (2020c)	58.56	[57.75-59.31]	7.99	[7.94-8.03]
	Justification	58.81	[58.22-59.41]	8.23	[7.93-8.04]
	PubHealth				
	Lead^{3}	44.44	[43.05-45.68]	9.12	[9.05-9.19]
	Lead^{5}	45.96	[44.80-46.98]	8.85	[8.79-8.91]
Supervised	Top^{5} (Supervised)	48.63	[47.91-49.44]	8.67	[8.62-8.72]
	Top^{5}+Edits^{5}	53.79	[53.01-54.56]	8.37	[8.31-8.42]
	Top^{5}+Edits^{5}+Par	61.39	[60.71-62.10]	7.97	[7.92-8.03]
Unsupervised	Top^{5} (Unsupervised)	45.20	[44.41-46.04]	8.94	[8.89-8.98]
	Top^{5}+Edits^{5}	50.74	[49.89-51.53]	8.63	[8.57-8.68]
	Top^{5}+Edits^{5}+Par	60.07	[59.37-60.77]	8.15	[8.09-8.20]
	MTSum^{3} (2020c)	48.73	[47.81-49.66]	8.88	[8.82-8.94]
	Justification	49.29	[48.27-50.40]	9.17	[9.08-9.26]

Table 5.9: Readability measures (§5.5.1) over the test splits. Readability measures include 95% confidence intervals based on 1000 random re-samples from the corresponding split (§5.5.1, Metrics.). We report results reported in MTSum, where we have the outputs to compute readability. We also report results given the gold explanation – Justification. Readability scores for Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$ and Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$+Para are statistically significant (p<0.05) compared to Top$^{\mathrm{N}}$, and to MTSum, except for the score in purple.

correct according to the ROUGE scores. For example, in the second instance, "Death rates do not define an epidemic" in the post-edited explanation and "Death rates alone don't determine whether an outbreak is an epidemic" from the original explanation express the same meaning, but contain both paraphrases and words without added meaning that would decrease the final ROUGE scores. Finally, compared to the original explanation, the post-edited explanations for both instances have preserved the information needed for fact checking.

	Method	R-1↗	R-1 CI	R-2↗	R-2 CI	R-L↗	R-L CI
				LIAR-PLUS			
Baselines	Lead^{4}	*28.11*	[27.39-28.39]	*6.96*	[6.52 - 7.33]	*24.38*	[23.73-24.68]
	Lead^{6}	29.15	[28.66-29.69]	8.28	[7.85 - 8.67]	25.84	[25.35-26.30]
Sup.	Top^{6} (Supervised)	34.42	[33.78-35.00]	12.36	[11.85-12.84]	30.58	[30.01-31.13]
	Top^{6}+Edits^{6}	33.92	[33.31-34.53]	11.73	[11.29-12.24]	30.01	[29.43-30.60]
	Top^{6}+Edits^{6}+Par	33.94	[33.37-34.47]	11.25	[10.81-11.73]	30.08	[29.49-30.59]
Unsup.	Top^{6} (Unsupervised)	29.63	[29.03-30.07]	7.58	[7.53 - 8.25]	25.86	[25.52-26.47]
	Top^{6}+Edits^{6}	28.93	[28.42-29.42]	7.06	[6.74 - 7.43]	25.14	[24.69-25.61]
	Top^{6}+Edits^{6}+Par	28.98	[28.50-29.52]	6.84	[6.51 - 7.16]	25.39	[24.95-25.87]
	MTSum^{4} (2020c)	*35.70*	[34.23-35.39]	*13.51*	[12.47-13.67]	*31.58*	[30.07-31.21]
				PubHealth			
Baselines	Lead^{3}	*29.01*	-	*10.24*	-	*24.18*	-
	Lead^{3}	23.05	[22.53-23.59]	6.28	[5.48 - 6.37]	19.27	[18.42-19.40]
	Lead^{5}	23.73	[22.50-23.62]	6.86	[5.81 - 6.58]	20.67	[19.08-20.18]
Sup.	Top^{5} (Supervised)	29.93	[28.87-30.97]	12.42	[11.44-13.63]	26.24	[25.21-27.44]
	Top^{5}+Edits^{5}	29.38	[28.45-30.33]	11.16	[10.17-12.15]	25.41	[24.48-26.41]
	Top^{5}+Edits^{5}+Par	28.40	[27.55-29.17]	9.56	[8.89 - 10.23]	24.37	[23.52-25.10]
Unsup.	Top^{5} (Unsupervised)	23.52	[22.95-24.12]	6.12	[5.76 - 6.46]	19.93	[19.37-20.45]
	Top^{5}+Edits^{5}	23.09	[22.55-23.64]	5.56	[5.24 - 5.92]	19.44	[18.93-19.93]
	Top^{5}+Edits^{5}+Par	23.35	[22.85-23.86]	5.38	[5.09 - 5.71]	19.56	[19.08-20.03]
	AbstrSum^{3} (2020a)	*32.30*	-	*13.46*	-	*26.99*	-
	MTSum^{3} (2020c)	33.55	[29.79-31.65]	13.12	[11.17-13.42]	29.41	[25.27-27.31]

Table 5.10: ROUGE-1/2/L F_1 scores (§5.5.2) of supervised (Sup.) and usupervised (Unsup.) methods over the test splits. In *italics*, we report results reported from prior work, where we do not always have the outputs to compute the confidence intervals. Underlined ROUGE scores of the Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$ and Top$^{\mathrm{N}}$+Edits$^{\mathrm{N}}$+Para are statistically significant ($p < 0.05$) compared to the input Top$^{\mathrm{N}}$ ROUGE scores, N={5,6}.

	Method	Flesch ↗	Flesch CI	DC ↘	DC CI
	LIAR-PLUS				
Baselines	Lead^{4}	50.89	[50.01-51.63]	8.75	[8.71-8.80]
	Lead^{6}	53.01	[52.41-53.64]	8.43	[8.39-8.47]
Supervised	Top^{6} (Supervised)	57.77	[57.15-58.38]	7.91	[7.87-7.95]
	Top^{6}+Par	63.88	[63.31-64.45]	7.55	[7.51-7.59]
	Top^{6}+Edits$^{\text{IE}}$	55.70	[55.03-56.36]	6.53	[6.50-6.56]
	Top^{6}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$	59.52	[58.89-60.17]	7.78	[7.73-7.83]
	Top^{6}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$+Par	66.05	[65.53-66.61]	7.46	[7.41-7.50]
Unsupervised	Top^{6} (Unsupervised)	52.84	[52.27-53.36]	8.52	[8.48-8.55]
	Top^{6}+Par	50.92	[50.18-51.58]	6.97	[6.94-7.01]
	Top^{6}+Edits$^{\text{IE}}$	50.70	[50.13-51.27]	6.92	[6.89-6.94]
	Top^{6}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$	54.76	[54.15-55.34]	8.39	[8.34-8.43]
	Top^{6}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$+Par	61.80	[61.17-62.42]	8.01	[7.97-8.05]
	MTSum^{4} (2020c)	58.08	[57.33-58.83]	8.03	[7.97-8.08]
	Justification	58.90	[58.23-59.68]	8.26	[8.20-8.32]
	LIAR-PLUS Test Split Ablation				
Supervised	Top^{6}+Par	64.45	[63.81-65.04]	7.52	[7.48-7.56]
	Top^{6}+Edits$^{\text{IE}}$	56.26	[55.37-57.04]	6.51	[6.48-6.55]
Unsupervised	Top^{6}+Par	59.83	[59.20-60.36]	8.21	[8.16-8.25]
	Top^{6}+Edits$^{\text{IE}}$	59.34	[58.72-59.91]	8.14	[8.10-8.18]
	PubHealth				
	Lead^{3}	44.76	[43.49-45.87]	9.12	[9.05-9.20]
	Lead^{5}	46.00	[44.80-46.92]	8.88	[8.83-8.94]
Supervised	Top^{5} (Supervised)	49.56	[48.73-50.27]	8.63	[8.58-8.68]
	Top^{5}+Par	47.38	[46.50-48.15]	7.07	[7.04-7.12]
	Top^{5}+Edits$^{\text{IE}}$	57.53	[56.85-58.19]	8.18	[8.13-8.24]
	Top^{5}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$	54.30	[53.58-54.97]	8.33	[8.27-8.38]
	Top^{5}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$+Par	61.51	[60.89-62.19]	7.96	[7.91-8.01]
Unsupervised	Top^{5} (Unsupervised)	43.55	[42.51-44.52]	9.26	[9.19-9.32]
	Top^{5}+Par	42.70	[41.60-43.59]	7.35	[7.31-7.40]
	Top^{5}+Edits$^{\text{IE}}$	56.33	[55.68-56.97]	8.35	[8.31-8.40]
	Top^{5}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$	50.46	[49.62-51.23]	8.65	[8.59-8.70]
	Top^{5}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$+Par	60.25	[59.56-60.89]	8.13	[8.08-8.19]
	MTSum^{3} (2020c)	49.69	[48.73-50.53]	8.81	[8.75-8.88]
	Justification	48.20	[47.25-49.16]	9.22	[9.15-9.32]
	PubHealth - Test Split Ablation				
Supervised	Top^{5}+Par	46.23	[45.33-47.07]	7.11	[7.07-7.15]
	Top^{5}+Edits$^{\text{IE}}$	57.29	[56.58-57.96]	8.21	[8.16-8.26]
Unsupervised	Top^{5}+Par	42.30	[41.34-43.28]	7.36	[7.31-7.41]
	Top^{5}+Edits$^{\text{IE}}$	55.79	[55.16-56.44]	8.39	[8.34-8.43]

Table 5.11: Readability measures (§5.5.1) over the validation splits. Readability measures include 95% confidence intervals (§5.5.1, Metrics.). We report results from prior work – MTSum, where we have the outputs to compute readability.

	Method	R-1↗	R-1 CI	R-2↗	R-2 CI	R-L↗	R-L CI
			LIAR-PLUS				
Base.	Lead^{4}	27.52	[26.99-28.00]	6.90	[6.54-7.30]	24.01	[23.53-24.49]
	Lead^{6}	28.93	[28.39-29.43]	8.32	[7.92-8.76]	25.67	[25.15-26.16]
Sup.	Top^{6} (Sup.)	34.35	[33.71-34.94]	12.20	[11.72-12.70]	30.51	[29.97-31.09]
	Top^{6}+Par	34.51	[33.90-35.08]	11.49	[11.04-11.96]	30.68	[30.15-31.25]
	Top^{6}+Edits$^{\text{IE}}$	25.18	[24.63-25.72]	8.60	[8.23-8.98]	22.08	[21.58-22.55]
	Top^{6}+Edits$^{\text{IE}}$+Edits$^{\text{Gram}}$	34.07	[33.45-34.71]	11.58	[11.14-12.05]	30.12	[29.55-30.70]
	Top^{6}+Edits^{6}+Par	34.20	[33.62-34.78]	11.05	[10.59-11.46]	30.26	[29.71-30.83]
Uns.	Top^{6} (Uns.)	29.29	[28.79-29.78]	7.99	[7.64-8.37]	25.84	[25.36-26.30]
	Top^{6}+Par	22.74	[22.31-23.24]	5.56	[5.29-5.82]	19.50	[19.06-19.93]
	Top^{6}+Edits$^{\text{IE}}$	21.46	[20.98-21.93]	5.67	[5.43-5.93]	18.76	[18.34-19.20]
	Top^{6}+Edits^{6}+Edits$^{\text{Gram}}$	29.02	[28.50-29.53]	7.47	[7.10-7.83]	25.52	[25.02-25.97]
	Top^{6}+Edits^{6}+Par	29.41	[28.89-29.90]	7.26	[6.90-7.60]	25.90	[25.43-26.37]
	MTSum^{4} (2020c)	34.80	[34.13-35.39]	12.87	[12.29-13.46]	30.66	[30.08-31.26]
			LIAR-PLUS Test Split Ablation				
Sup.	Top^{6}+Para	34.62	[34.02-35.23]	11.79	[11.33-12.21]	30.81	[30.26-31.35]
	Top^{6}+Edits$^{\text{IE}}$	25.48	[25.00-26.03]	8.75	[8.41-9.09]	22.29	[21.78-22.79]
Uns.	Top^{6}+Para	34.61	[34.08-35.19]	11.77	[11.30-12.26]	30.81	[30.20-31.38]
	Top^{6}+Edits$^{\text{IE}}$	25.48	[24.87-26.01]	8.75	[8.41-9.13]	22.29	[21.81-22.76]
			PubHealth				
Base.	Lead^{3}	23.18	[22.69-23.71]	5.55	[5.14-6.00]	18.74	[18.31-19.19]
	Lead^{5}	23.31	[22.71-23.90]	6.07	[5.70-6.46]	19.60	[19.04-20.14]
Sup.	Top^{5} (Sup.)	30.37	[29.40-31.42]	12.64	[11.51-13.77]	26.46	[25.42-27.49]
	Top^{5}+Par	22.49	[21.62-23.39]	8.96	[8.22-9.75]	19.73	[18.82-20.64]
	Top^{5}+Edits$^{\text{IE}}$	29.76	[28.81-30.70]	10.75	[9.91-11.63]	25.44	[24.56-26.32]
	Top^{5}+Edits^{5}+Edits$^{\text{Gram}}$	29.62	[28.69-30.49]	11.20	[10.31-12.19]	25.54	[24.62-26.46]
	Top^{5}+Edits^{5}+Par	28.81	[27.97-29.70]	9.67	[8.94-10.37]	24.47	[23.71-25.31]
Uns.	Top^{5} (Uns.)	23.80	[23.27-24.31]	5.76	[5.36-6.13]	19.24	[18.74-19.70]
	Top^{5}+Par	18.28	[17.64-18.88]	4.50	[4.22-4.79]	15.50	[14.90-16.14]
	Top^{5}+Edits$^{\text{IE}}$	24.44	[23.85-25.04]	5.97	[5.67-6.31]	20.51	[19.98-21.01]
	Top^{5}+Edits^{5}+Edits$^{\text{Gram}}$	23.74	[23.20-24.28]	5.72	[5.41-6.03]	19.76	[19.27-20.30]
	Top^{5}+Edits^{5}+Par	23.96	[23.49-24.49]	5.46	[5.17-5.78]	19.98	[19.52-20.45]
	MTSum^{3} (2020c)	31.05	[30.08-32.09]	12.66	[11.44-13.82]	26.45	[25.46-27.50]
			PubHealth Test Split Ablation				
Sup.	Top^{5}+Par	22.09	[21.24-22.95]	8.75	[7.97-9.54]	19.48	[18.60-20.36]
	Top^{5}+Edits$^{\text{IE}}$	29.46	[28.57-30.37]	10.71	[9.86-11.63]	25.53	[24.67-26.40]
Uns.	Top^{5}+Par	18.11	[17.44-18.77]	4.41	[4.16-4.70]	15.48	[14.92-16.10]
	Top^{5}+Edits$^{\text{IE}}$	18.11	[17.44-18.77]	4.41	[4.16-4.70]	15.48	[14.92-16.10]

Table 5.12: ROUGE-1/2/L F_1 scores (§5.5.2) of baselines (Base.), supervised (Sup.) and usupervised (Uns.) methods over the validation splits. In *italics*, we report results reported from prior work, where we do not always have the outputs to compute the confidence intervals.

Top^5**:** Heavily-armed Muslims shouting “Allahu Akbar” open fire campers and hikers in a park. A heavily armed group of Middle Eastern looking Muslim men was arrested outside Los Angeles after opening fire upon hikers and campers in a large State Park in the area . There was no evidence found that a crime had been committed by any of the subjects who were detained and they were released . Also, the police report described the men only as “ males ,” not “Middle Eastern males ” or “Muslim males .” The website that started this rumor was Superstation95, which is not a “superstation” at all but rather a repository of misinformation from Hal Turner, who in 2010 was sentenced to 33 months in prison for making death threats against three federal judges. No credible news reports made any mention of the “Allahu Akbar” claim, and no witnesses stated they had been “shot at” by the men while hiking or camping.
Top^5+Edits^5**:** Heavily-armed Muslims **males** shouting “Allahu Akbar” open fire in a park. A heavily armed group of Middle Eastern looking Muslim men was arrested after opening fire upon hikers and campers in a large State Park **outside Los Angeles**. There was no evidence found that a crime had been committed by any of the subjects on **campers and hikers** . Also, the police report described the men only as “,” not “Middle Eastern” or “Muslim.” The website that started this rumor was Superstation95, which is not a “superstation” at all but rather a repository of misinformation from Hal Turner, who in 2010 was sentenced to 33 months in prison for making death threats against three federal judges . No credible news reports made any mention of the “Allahu Akbar” claim, and no witnesses stated they had been “shot at”.
Top^5+Edits^5+Par **Muslims shout** "Allahu Akbar" open fire in a park. A heavily armed group of Middle Eastern looking Muslim men was arrested after opening fire on hikers and campers in a large State Park outside Los Angeles. There was no evidence that a crime had been committed by **any of the campers or hikers**. The website that started this rumor was Superstation95, which is not a “superstation” at all but rather a repository of misinformation from Hal Turner, who in 2010 was sentenced to 33 months in prison. **There were no credible news reports that mentioned the Allahu Akbar claim, and no witnesses that said they had been shot at.**
Original Explanation: Secondary reporting claiming that Muslim men fired upon hikers (and that the media covered it up) appeared on a site that had previously inaccurately claimed Illinois had applied Sharia law to driver’s licenses, that Target introduced “Sharia-compliant” checkout lanes, and that Muslims successfully banned Halloween at a New Jersey school.
Claim: The media covered up an incident in San Bernardino during which several Muslim men fired upon a number of Californian hikers. **Label:** False
Top^5: The article claims the CDC might have to stop calling COVID-19 an epidemic because the death rate is becoming so low that it wouldn’t meet the CDC’s definition of epidemic. The latest CDC statement made public when the Facebook post was made said deaths attributed to COVID-19 decreased from the previous week, but remained at the epidemic threshold, and were likely to increase . Moreover death rates alone do not define an epidemic. Amid news headlines that the United States set a daily record for the number of new coronavirus cases, an article widely shared on Facebook made a contrarian claim . The CDC page says: “Epidemic refers to an increase, often sudden, in the number of cases of a disease above what is normally expected in that population in that area.”
Top^5+Edits^5**:** The article claims the CDC might have to stop calling COVID-19 an epidemic **in that population** because the death rate is becoming so low that it wouldn’t meet the CDC’s definition of epidemic. The latest CDC statement an article from the previous week said deaths decreased, but. Moreover, death rates do not define an epidemic. Amid news headlines that the United States **on Facebook** set a daily record for the number of new coronavirus cases, The CDC page when the Facebook post was made says: “Epidemic refers to an increase, often sudden, in the number of cases of a disease attributed to COVID-19.”
Top^5+Edits^5+Par**:** **According to the article**, the CDC might have to stop calling COVID-19 an epidemic because the death rate is so low that it wouldn’t meet **their** definition of an epidemic. **An article from the previous week said deaths decreased, but that’s what the latest CDC statement says.** Death rates do not define an epidemic. The CDC’s page on Facebook says Epidemic refers to an increase, often sudden, in the number of cases of a disease attributed to COVID-19.
Original Explanation: Despite a dip in death rates, which are expected to rise again, the federal Centers for Disease Control and Prevention still considers COVID-19 an epidemic. Death rates alone don’t determine whether an outbreak is an epidemic.
Claim: The CDC may have to stop calling COVID-19 an ‘epidemic’ due to a remarkably low death rate. **Label:** False

Table 5.13: Example explanations – extracted Top^5 RCs, the iterative editing, and the latter with paraphrasing on top, taken from the test split of PubHealth. Each color designates an edit operation – **reordering**, **deletion**, and **paraphrasing**. The underlining designates the position in the text where the corresponding operation will be applied in the next step – post-editing and paraphrasing.

6 Multi-Hop Fact Checking of Political Claims

6.1 Introduction

Recent progress in machine learning has seen interest in automating complex reasoning, where a conclusion can be reached only after following logically connected arguments. To this end, multi-hop datasets and models have been introduced, which learn to combine information from several sentences to arrive at an answer. While most of them concentrate on question answering, fact checking is another task that often requires a combination of multiple evidence pieces to predict a claim's veracity.

Existing fact checking models usually optimize only the veracity prediction objective and assume that the task requires a single inference step. Such models ignore that often several linked evidence chunks have to be explicitly retrieved and combined to make the correct veracity prediction. Moreover, they do not provide explanations of their decision-making, which is an essential part of fact checking.

Atanasova et al. (2020b) note the importance of providing explanations for fact checking verdicts, and propose an extractive summarization model, which optimizes a ROUGE score metric w.r.t. a gold explanation. Gold explanations for this are obtained from the LIAR-PLUS (Alhindi et al., 2018) dataset, which is constructed from PolitiFact[1] articles written by professional fact checking journalists. However, the dataset does not provide guidance on the several relevant evidence pieces that have to be linked and assume that the explanation requires a single reasoning step. FEVER (Thorne et al., 2018) is another fact checking dataset, which contains annotations of evidence sentences from Wikipedia pages. However, it consists of manually augmented claims, which require limited reasoning capabilities for verification as the evidence mostly consists of one or two sentences.

To provide guidance for the multi-hop reasoning process of a claim's verification and facilitate progress on explainable fact checking, we introduce `PolitiHop`, a dataset of 500 real-world claims with manual annotations of sets of interlinked evidence chunks from PolitiFact articles needed to predict the claims' labels. We provide insights from the annotation process, indicating that fact checking real-world claims is an elaborate process requiring multiple hops over evidence chunks, where multiple evidence sets are also possible.

[1] https://www.politifact.com/

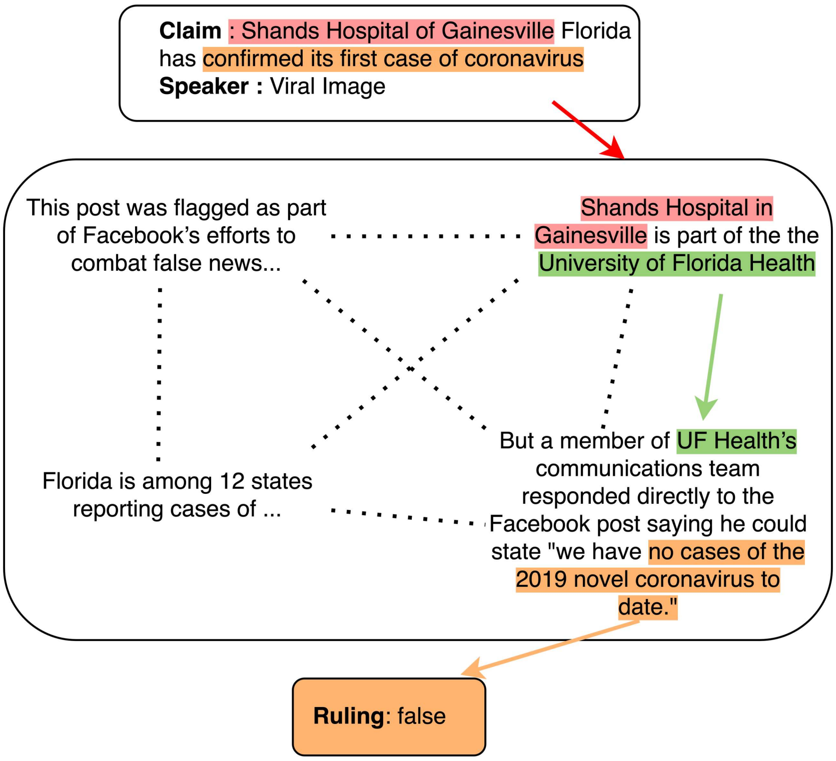

Claim : Shands Hospital of Gainesville Florida has confirmed its first case of coronavirus
Speaker : Viral Image
This post was flagged as part of Facebook's efforts to combat false news...
Shands Hospital in Gainesville is part of the the University of Florida Health
Florida is among 12 states reporting cases of ...
But a member of UF Health's communications team responded directly to the Facebook post saying he could state "we have no cases of the 2019 novel coronavirus to date."
Ruling: false

Figure 6.1: An illustration of multiple hops over an instance from PolitiHop. Each instance consists of a claim, a speaker, a veracity label, and a PolitiFact article the annotated evidence sentences. The highlighted sentences represent the evidence sentences a model needs to connect to arrive at the correct veracity prediction.

To assess the difficulty of the task, we conduct experiments with lexical baselines, as well as a single-inference step model – BERT (Devlin et al., 2019), and a multi-hop model – Transformer-XH (Zhao et al., 2020). Transformer-XH allows for the sharing of information between sentences located anywhere in the document by eXtra Hop attention and achieves the best performance. We further study whether multi-hop reasoning learned with Transformer-XH can be transferred to PolitiHop. We find that the model cannot leverage any reasoning skills from training on FEVER, while training on LIAR-PLUS improves the performance on PolitiHop. We hypothesize that this is partly due to a domain discrepancy, as FEVER is constructed from Wikipedia and consists of claims requiring only one or two hops for verification. In contrast, LIAR-PLUS is based on PolitiFact, same as PolitiHop.

Finally, we perform a detailed error analysis to understand the models' shortcomings and recognize possible areas for improvement. We find that the models perform worse when the gold evidence sets are larger and that, surprisingly, named entity (NE) overlap between evidence and non-evidence sentences does not have a negative effect on either evidence retrieval or label prediction. The best results for Transformer-XH on the dev and test sets are for a different number of hops – 2 and 6, indicating that having a fixed parameter for the number of hops is a downside of Transformer-XH; this should instead be learned for each claim. Overall, our experiments constitute a solid basis to be used for future developments.

To summarise, our **contributions** are as follows:

- We document the first study on multi-hop fact checking of political claims.
- We create a dataset, `PolitiHop`, for the task.
- We study whether reasoning skills learned with a multi-hop model on similar datasets can be transferred to `PolitiHop`.
- We analyze to what degree existing multi-hop reasoning methods are suitable for the task.

6.2 Multi-Hop Fact Checking

A multi-hop fact checking model $f(X)$ receives as an input $X = \{(claim_i, document_i) |i \in [1, |X|]\}$, where $document_i = [sentence_{ij}|j \in [1, |document_i|]]$ is the corresponding PolitiFact article for $claim_i$ and consists of a list of sentences. During the training process, the model learns to (i) select which sentences from the input contain evidence needed for the veracity prediction $y_i^S = [y_{ij}^S \in \{0, 1\}|j \in [1, |document_i|]]$ (**sentence selection task**), where 1 indicates that the sentence is selected as an evidence sentence; and (ii) predict the veracity label of the claim $y_i^L \in \{True, False, Half-True\}$, based on the extracted evidence (**veracity prediction task**). The sentences selected by the model as evidence provide *sufficient* explanation, which allows to verify the corresponding claim *efficiently* instead of reading the whole article. Each evidence set consists of k sentences, where $k \in [1, max_{i \in [1,|X|]}(|document_i|)]$ is a hyper-parameter of the model. Figure 6.1 illustrates the process of multi-hop fact checking, where multiple evidence sentences provide different information, which needs to be connected in logical order to reach the final veracity verdict.

6.2.1 Dataset

We present `PolitiHop`, the first dataset for multi-hop fact checking of real-world claims. It consists of 500 manually annotated claims in written English, split into a training (300 instances) and a test set (200 instances). For each claim, the corresponding PolitiFact article was retrieved, which consists of a discussion of each claim and its veracity, written by a professional fact checker. The annotators then selected sufficient sets of evidence sentences from said articles. As sometimes more than one set can be found to describe a reason behind the veracity of a claim independently, we further take each set in the training split as a separate instance, resulting in 733

Statistic	Test	Train
#Words per article	569 (280.8)	573 (269.1)
#Sent. per article	28 (12.8)	28 (12.8)
#Evidence sent. per article	11.75 (5.56)	6.33 (2.98)
#Evidence sent. per set	2.88 (1.43)	2.59 (1.51)
#Sets per article	4.08 (1.83)	2.44 (1.28)
Label Distribution		
False	149	216
Half-true	30	47
True	21	37

Table 6.1: `PolitiHop` dataset statistics. Test set statistics are calculated for a union of two annotators; train instances are annotated by one annotator only, which makes some measures different across splits. We report the mean and standard deviation (in parentheses).

training examples. Each training example is annotated by one annotator, whereas each test example is annotated by two. We split the training data into train and dev datasets, where the former has 592 examples and the latter – 141. For veracity prediction, we arrived at Krippendorf's α and Fleiss' κ agreement values of 0.638 and 0.637, respectively. By comparison, Thorne et al. (2018) reported Fleiss' κ of 0.684 on FEVER. For the sentence prediction, we attain Krippendorf's α of 0.437. A more in-depth description of the annotation process can be found in the appendix.

Table 6.1 presents statistics of the dataset. The average number of evidence sentences per set is above 2, which already indicates that the task is more complex than the FEVER dataset. In FEVER, 83.2% of the claims require one sentence, whereas in `PolitiHop`, only 24.8% require one sentence.

6.3 Models

We compare the performance of five different models to measure the difficulty of automating the task.

Majority. Label prediction only. The majority baseline labels all claims as false.

Random. We pick a random number $k \in [1, 10]$ and then randomly choose k sentences from the document as evidence. For label prediction, we randomly pick one of the labels.

TF-IDF. For each instance x_i we construct a vector $v_i^C = [v_{il}^C | l \in [0, |N^C|]]$ with TF-IDF scores v_{il}^C for all n-grams N^C found in all of the claims; and one vector $v_i^D = [v_{im}^D | m \in [0, |N^D|]]$ with TF-IDF scores v_{im}^D for all n-grams N^D found in all of the documents, where $n \in [2, 3]$. We then train a Naive Bayes model $g(V)$, where $V = \{v_i = (v_i^C \cdot v_i^D) | i \in [0, |X|]\}$ is the concatenation of the two feature vectors.

BERT. We first train a Transformer model (Vaswani et al., 2017), which does not include a multi-hop mechanism, but applies a single inference step to both the evidence retrieval and the label prediction tasks. We employ BERT (Devlin et al., 2019) with the base pre-trained weights. Each sentence from a fact checking document is encoded separately, combined with the claim and the author of the claim. We refer to the encoded triple as *node* τ. The tokens of one node $x_\tau = \{x_{\tau,j} | j \in [1, |x_\tau|]\}$ are encoded with the BERT model into contextualized distributed representations: $h_\tau = \{h_{\tau,j} | j \in [1, |x_\tau|]\}$. The encoded representations of all nodes are passed through two feed-forward layers:

$$p(y^L|\tau) = softmax(Linear(h_{\tau,0})) \tag{6.1}$$

$$p(y^S|\tau) = softmax(Linear(h_{\tau,0})) \tag{6.2}$$

$$p(y^L|X) = \sum_{\tau} p(y^L|\tau) p(y^S|\tau) \tag{6.3}$$

The first layer predicts the veracity of the claim given a particular node τ by using the contextual representation of the "[CLS]" token, located at the first position (Eq. 6.1). The second feed-forward layer learns the importance of each node in the graph (Eq. 6.2). The outputs of these two layers are combined for the final label prediction (Eq. 6.3). For evidence prediction, we choose k most important sentences, as ranked by the second linear layer. In our experiments, we set $k = 6$ since this is the average number of evidence sentences selected by a single annotator. The implementation of the feed-forward prediction layers is the same as in Transformer-XH, described below, and can be viewed as an ablation of Transformer-XH removing the eXtra Hop attention layers.

Transformer-XH. Transformer-XH is a good candidate for a multi-hop model for our setup as it has previously achieved the best multi-hop evidence retrieval results on FEVER. It is also inspired by and improves over other multi-hop architectures (Liu et al., 2020c; Zhou et al., 2019), and we conjecture that the results should be generalisable for its predecessors as well. Not least, its architecture allows for ablation studies of the multi-hop mechanism. Following previous work on applying Transformer-XH to FEVER (Zhao et al., 2020), we encode node representations as with the BERT model and construct a fully connected graph with them. Transformer-XH uses eXtra hop attention layers to enable information sharing between the nodes. An eXtra hop attention layer is a Graph Attention layer (GAT) (Veličković et al., 2018), which receives as input a graph $\{X, E\}$ of all evidence nodes X and the edges between them E, where the edges encode the attention between two nodes in the graph. Each eXtra hop layer computes the attention between a node and its neighbors, which corresponds to one hop of reasoning across nodes. Transformer-XH applies L eXtra hop layers to the BERT node encodings H_0, which results in new representations H_L that encode the information shared between the nodes, unlike BERT, which encodes each input sentence separately. We use three eXtra hop layers as in (Zhao et al., 2020), which corresponds to three-hop reasoning, and we experiment with varying the number of hops. The representations H_L are passed to the final two linear layers for label and evidence prediction as in BERT. The final prediction of the veracity label $p(y^L|\{X, E\})$ now can also leverage information exchanged in multiple hops between the nodes through the edges E between them.

6.4 Experiments

We address the following research questions:

- Can multi-hop architectures successfully reason over evidence sets on PolitiHop?
- How do multi-hop vs. single inference architectures fare in an adversarial evaluation, where named entities (NE) in evidence and non-evidence sentences overlap?
- Does pre-training on related small in-domain or large out-of-domain datasets improve model performance?

We further perform ablation studies to investigate the influence of different factors on performance (see Section 6.6).

6.4.1 Experimental Setup

Metrics. We use macro F_1 score and accuracy for the veracity prediction task and F_1 and precision for the evidence retrieval task. To calculate the performance on both tasks jointly, we use the FEVER score (Thorne et al., 2018), where the model has to retrieve at least one full evidence set and predict the veracity label correctly for the label prediction to count as correct. We consider a single evidence set to be sufficient for correct label prediction. As each example from train and dev sets in PolitiHop, and every example from LIAR-PLUS, has one evidence set, all evidence sentences need to be retrieved for these. The employed measures for evidence retrieval allow for comparison to related work and for relaxing the requirements on the models. We consider the FEVER score to be the best for evaluating explainable fact checking.

Dataset settings. We consider three settings: adversarial, full article, and even split. For full, the whole article for each claim is given as input. For even split, we pick all sentences from the same article, but restrict the number of non-evidence sentences to be at most equal to the number of evidence sentences. Non-evidence sentences are picked randomly. This results in a roughly even split between evidence and non-evidence sentences for the test set. Since we divide train and dev datasets into one evidence set per example, but keep all non-evidence for each, the number of non-evidence sentences for instances in these splits is usually 2-3 times larger than the number of evidence sentences. To examine if the investigated multi-hop models overfit on named entity (NE) overlaps, we further construct an `adversarial` dataset from the even split dataset by changing each non-evidence sentence to a random sentence from any PolitiFact article, which contains at least one NE present in the original evidence sentences. While such sentences can share information about a relevant NE, they are irrelevant for the claim. We argue that this is a good testbed to understand if a fact checking model can successfully reason over evidence sets and identify non-evidence sentences, even if they contain relevant NEs, which are rather surface features not indicating whether the sentence is relevant to the claim.

Training settings. We perform transfer learning, training on in-domain data (LIAR-PLUS, `PolitiHop`), out-of-domain data (FEVER), or a combination thereof. See the appendix for details on training regimes and hyper-parameters.

Note that the measures do not consider the order of the sentences in the evidence set, and the systems do not predict that as well. We believe other measures and models that take that into account should be explored in future work. Here we consider them to appear in the same order as in the document. This also corresponds to the way they were annotated.

6.5 Results

Full article setting. From the results in Table 6.2, we can observe that both BERT and Transformer-XH greatly outperform the Random and TF-IDF baselines. Out of BERT and Transformer-XH, neither model clearly outperforms the other on our dataset. This is surprising as Transformer-XH outperforms the BERT baselines by a significant margin on both FEVER and the multi-hop dataset HotpotQA (Zhao et al., 2020). However, we observe that the best performance is achieved with Transformer-XH trained on LIAR-PLUS, then fine-tuned on `PolitiHop`. It also achieves the highest FEVER scores on `PolitiHop` in that setting. Further, very low FEVER scores of both Transformer-XH and BERT indicate how challenging it is to retrieve the whole evidence set.

Adversarial setting. We train the Transformer-XH models on the `even split` setting, then evaluate it on both `adversarial` and `even split` datasets (see Table 6.3). The

	Dev					Test				
	L-F_1	L-Acc	E-F_1	E-Prec	FEVER	L-F_1	L-Acc	E-F_1	E-Prec	FEVER
Random	34.1	38.5	22.9	30.2	4.5	24.2	27.7	14.7	12.2	0.7
Majority	27.3	69.5	-	-	-	28.6	75.0	-	-	-
Annotator	-	-	-	-	-	76.3	-	52.4	49.2	-
TF-IDF	34.4	69.5	-	-	-	34.0	76.0	-	-	-
LIAR-PLUS full articles dataset										
BERT	45.4	70.9	**18.4**	**13.7**	**14.9**	**57.0**	76.0	**32.9**	**38.9**	**13.0**
Transformer-XH	**56.2**	**74.5**	17.1	12.8	14.2	56.3	**79.5**	30.3	35.8	12.0
PolitiHop full articles dataset										
BERT	54.7	69.5	**32.0**	**23.6**	31.9	**44.8**	**76.0**	**47.0**	**54.2**	**24.5**
Transformer-XH	**61.1**	**76.6**	30.4	22.3	**34.8**	43.3	75.5	44.7	51.7	23.5
LIAR-PLUS and PolitiHop full articles										
BERT	64.4	75.9	29.6	21.7	28.4	**57.8**	79.5	45.1	52.2	23.5
Transformer-XH	**64.6**	**78.7**	**32.4**	**23.8**	**38.3**	57.3	**80.5**	**47.2**	**54.5**	**24.5**

Table 6.2: PolitiHop results for label (L), evidence (E) and joint (FEVER) performance in the full setting. Best results with a particular training dataset (LIAR-PLUS/PolitiHop/LIAR-PLUS and PolitiHop) are emboldened and the best results across all set-ups are underlined.

	Dev					Test				
	L-F_1	L-Acc	E-F_1	E-Prec	FEVER	L-F_1	L-Acc	E-F_1	E-Prec	FEVER
even split	**58.1**	**71.6**	47.7	35.5	**52.5**	**62.9**	**82.0**	58.2	66.7	31.0
adversarial	56.5	70.9	**49.9**	**38.7**	46.8	56.4	77.0	**63.6**	**76.0**	**33.5**

Table 6.3: PolitiHop adversarial vs even split dataset results for label (L), evidence (E) and joint (FEVER) performance for Transformer-XH trained on LIAR-PLUS and PolitiHop on the even split setting. Best result emboldened.

model performs similarly in both settings. When compared on test sets, it achieves a higher FEVER score on the adversarial, but dev sets FEVER score is higher on the even split setting. Overall, the results show Transformer-XH is robust towards NE overlap.

Out-of-domain pre-training on FEVER. In this experiment, we examine whether pre-training Transformer-XH on the large, but out-of-domain dataset FEVER, followed by fine-tuning on LIAR-PLUS, then on PolitiHop improves results on PolitiHop. As can be seen from Table 6.4, it does not have a positive effect on performance in the full setting, unlike pre-training on LIAR-PLUS. We hypothesize that the benefits of using a larger dataset are outweighed by the downsides of it being out-of-domain. We further quantify the domain differences between datasets. We use Jensen-Shannon divergence (Lin, 1991), commonly employed for this purpose (Ruder and Plank, 2017). The divergence between FEVER and PolitiHop is 0.278, while between LIAR-PLUS

	Dev					Test				
	L-F$_1$	**L-Acc**	**E-F$_1$**	**E-Prec**	**FEVER**	**L-F$_1$**	**L-Acc**	**E-F$_1$**	**E-Prec**	**FEVER**
FEVER+LIAR-PLUS +`PolitiHop`	48.6	70.2	30.5	22.2	32.6	**59.9**	**83.0**	45.1	52.7	21.5
LIAR-PLUS +`PolitiHop`	**64.6**	**78.7**	**32.4**	**23.8**	**38.3**	57.3	80.5	**47.2**	**54.5**	**24.5**

Table 6.4: `PolitiHop` `full` results for label (L), evidence (E) and joint (FEVER) performance for Transformer-XH trained on different datasets. Best model emboldened.

and `PolitiHop` is 0.063, which further corroborates our hypothesis. Another reason might be that `PolitiHop` has several times more input sentences compared to FEVER. Labelling difference might matter as well: FEVER uses 'true', 'false' and 'not enough info', while `PolitiHop` uses 'true', 'false' and 'half-true'.

6.6 Analysis and Discussion

In Section 6.5, we documented experimental results on multi-hop fact checking of political claims. Overall, we found that multi-hop training on Transformer-XH gives small improvements over BERT, that pre-training on in-domain data helps, and that Transformer-XH deals well with an adversarial test setting. Below, we aim to further understand the impact of modeling multi-hop reasoning explicitly with a number of ablation studies:

- How the evidence set size affects performance;
- Varying the hops' number in Transformer-XH;
- The impact of evidence set size on performance;
- How NE overlap affects performance;
- To what extent Transformer-XH pays attention to relevant evidence sentences.

Further ablation studies can be found in the appendix, namely on: impact of varying the number of evidence sentences on evidence retrieval; how to weigh the different loss functions (for label vs. evidence prediction); if providing supervision for evidence sentence positions impacts performance; and to what degree high label confidence is an indication of high performance.

Varying the number of hops in Transformer-XH. We train Transformer-XH with a varying number of hops to see if there is any pattern in how many hops result in the best performance. Zhao et al. (2020) perform a similar experiment and find that 3 hops are best, similar for 2-5 hops, while the decrease in performance is noticeable for

1 and 6 hops. We experiment with hops between 1 and 7 (see Table 6.5). Evidence retrieval performance is quite similar in each case. There are some differences for the label prediction task: 1 and 2 hops have slightly worse performance, the 4-hop model has the highest test score and the lowest dev score, while the exact opposite holds for the 5-hop model. Therefore, no clear pattern can be found. One reason for this could be the high variance of the annotated evidence sentences in `PolitiHop`.

	Dev					Test				
	L-F_1	L-Acc	E-F_1	E-Prec	FEVER	L-F_1	L-Acc	E-F_1	E-Prec	FEVER
1	54.0	**75.2**	47.1	35.2	52.5	58.9	79.5	58.7	**67.6**	33.0
2	56.1	73.0	47.5	35.4	**53.9**	59.8	78.5	58.1	66.7	32.5
3	58.1	71.6	**47.7**	**35.5**	52.5	62.9	82.0	58.2	66.7	31.0
4	53.3	70.9	47.0	34.9	50.4	**65.0**	**82.0**	**58.9**	**67.6**	33.0
5	**59.6**	73.0	**47.7**	**35.5**	51.8	55.3	76.5	58.7	67.3	32.0
6	56.5	73.0	45.9	34.2	50.4	64.9	81.5	57.5	66.0	**35.0**
7	56.3	71.6	46.4	34.6	50.4	62.8	81.5	57.9	66.4	33.0

Table 6.5: `PolitiHop` Transformer-XH results for label (L), evidence (E) and joint (FEVER) performance for training on the LIAR-PLUS + `PolitiHop even split` datasets with a varying number of hop layers. Best sentence number emboldened.

Evidence set size vs. performance. Not surprisingly, larger number of evidence sentences leads to higher precision and lower recall, resulting in a lower FEVER score. This is true for both models, as Table 6.6 (top) indicates. We also notice that the smaller the number, the smaller the ratio of evidence to non-evidence sentences. For instance, if a claim has two sets of evidence, one of size 1 and the other of size 3, then after splitting into one example per set, there are 4 non-evidence sentences in each of the two examples, but the one with set of size 1 has only one evidence sentence – which decreases the evidence to non-evidence ratio and makes it more difficult to achieve high precision.

Named entity overlap vs. performance. To measure the effect of having the same NEs in evidence and non-evidence sentences, we computed NE overlap – a measure of the degree to which evidence and non evidence sentences share NEs. We compute the overlap as $|E \cap N|/|E \cup N|$; E and N are sets of NEs in evidence and non-evidence sentences, respectively. Table 6.6 (bottom) shows that a higher NE overlap results in more confusion when retrieving evidence sentences, but it does not have a significant influence on label prediction in the case of Transformer-XH. For BERT, higher NE overlap leads to a bigger, negative effect on both tasks. This suggests Transformer-XH is more robust to NE overlaps.

Attention over evidence sentences. We investigate what attention patterns Transformer-XH learns. Ideally, attention flowing from evidence sentences should be

	Transformer-XH					BERT				
	L-F_1	L-Acc	E-F_1	E-Prec	FEVER	L-F_1	L-Acc	E-F_1	E-Prec	FEVER
1 or 2 evidence sentences	**63.9**	**76.8**	**43.7**	**29.2**	**74.4**	53.5	72.0	41.3	27.5	62.2
3+ evidence sentences	**60.9**	66.1	**67.5**	**56.2**	**42.4**	57.8	66.1	65.8	54.8	40.7
$<$ 40% NE overlap	**62.5**	**77.0**	**59.1**	**46.2**	**62.3**	62.0	75.4	57.7	45.1	59.0
$\geq$ 40% NE overlap	**63.6**	**71.0**	**48.5**	**35.5**	**60.9**	47.5	66.7	46.2	33.8	49.3

Table 6.6: PolitiHop adversarial dev set performance vs. (top) evidence set size and (bottom) NE overlap between evidence and non-evidence sentences for label (L), evidence (E) and joint (FEVER) performance. Better model emboldened.

ev → non-ev	ev → ev	non-ev → non-ev	non-ev → ev
1.085	1.076	0.966	0.964

Table 6.7: Attention weights in the last eXtra hop layer of Transformer-XH. The numbers are the average ratios of the actual attention weights to average attention weight of the given graph.

higher than from non-evidence ones since this determines how much they contribute to the final representations of each sentence. To do this, we inspect the weights in the final eXtra hop layer. We normalize results by measuring the ratio of the given attention to the average attention for the given graph: 1 means average, over/under 1 means more/less than average. Table 6.7 shows average ratios for evidence vs. non-evidence sentences. One notable finding is that attention weights from evidence sentences are higher than average, and attention from non-evidence sentences is lower. The Welch t-test indicates that the difference is significant with a p-value lower than 10^{-30}. So, attention weights get more importance on average, but the magnitude of this effect is quite limited. This shows the limitations of using Transformer-XH for this task.

6.7 Related Work

Fact checking. Several datasets have been released to assist in automating fact checking. Vlachos and Riedel (2014) present a dataset with 106 political claim-verdict pairs. The FakeNewsChallenge [2], provides 50K headline-article pairs and formulates the task of fact checking as stance detection between the headline and the body of the article. The relationship between these two tasks is further explored in Hardalov et al. (2021). Wang (2017) extract 12.8K claims from PolitiFact constituting the

[2] http://www.fakenewschallenge.org/

LIAR dataset. Alhindi et al. (2018) introduce the LIAR-PLUS dataset extending the latter with automatically extracted summaries from PolitiFact articles. These are, however, high-level explanations that omit evidence details. LIAR-PLUS also does not provide annotation of particular evidence sentences from the article leading to the final verdict and the possible different evidence sets. Augenstein et al. (2019) present a real-world dataset constructed from 26 fact checking portals, including PolitiFact, consisting of 35k claims paired with crawled evidence documents. Thorne et al. (2018) present the FEVER dataset, consisting of 185K claims produced by manually re-writing Wikipedia sentences. Furthermore, Niewinski et al. (2019) from the FEVER'2019 shared task (Thorne et al., 2019b) and Hidey et al. (2020) use adversarial attacks to show the vulnerability of models trained on the FEVER dataset to claims that require more than one inference step. Unlike prior work, we construct a dataset with annotations of the different reasoning sets and the multiple hops that constitute them.

Multi-hop datasets. Multi-hop reasoning has been mostly studied in the context of Question Answering (QA). Yang et al. (2018) introduce HotpotQA with Wikipedia-based question-answer pairs requiring reasoning over multiple documents and provide gold labels for sentences supporting the answer. Welbl et al. (2018) introduce MedHop and WikiHop datasets for reasoning over multiple documents. These are constructed using Wikipedia and DrugBank as Knowledge Bases (KB), and are limited to entities and relations existing in the KB. This, in turn, limits the type of questions that can be generated. TriviaQA (Joshi et al., 2017) and SearchQA (Dunn et al., 2017) contain multiple documents for question-answer pairs but have few examples where reasoning over multiple paragraphs from different documents is necessary.

Multi-hop models. Chen and Durrett (2019) observe that models without multi-hop reasoning are still able to perform well on a large portion of the test dataset. Hidey et al. (2020) employ a pointer-based architecture, which re-ranks documents related to a claim and jointly predicts the sequence of evidence sentences and their stance to the claim. Asai et al. (2020) sequentially extract paragraphs from the reasoning path conditioning on the documents extracted on the previous step. CogQA (Ding et al., 2019) detect spans and entities of interest and then run a BERT-based Graph Convolutional Network for ranking. Nie et al. (2019) perform semantic retrieval of relevant paragraphs followed by span prediction in the case of QA and 3-way classification for fact checking. Zhou et al. (2019), Liu et al. (2020c), and Zhao et al. (2020) model documents as a graph and apply attention networks across the nodes of the graph. We use Zhao et al. (2020)'s model due to its strong performance in multi-hop QA on the HotpotQA dataset, in evidence-based fact checking on FEVER, and to evaluate its performance on real-world claim evidence reasoning.

6.8 Conclusions

In this paper, we studied the novel task of multi-hop reasoning for fact checking of real-world political claims, which encompasses both evidence retrieval and claim veracity prediction. We presented `PolitiHop`, the first political fact checking dataset with annotated evidence sentences. We compared several models on `PolitiHop` and found that the multi-hop architecture Transformer-XH slightly outperforms BERT in most of the settings, especially in terms of evidence retrieval, where BERT is easily fooled by named entity overlaps between the claim and evidence sentences. The performance of Transformer-XH is further improved when retrieving more than two evidence sentences and the number of hops larger than one, which corroborates the assumption of the multi-hop nature of the task.

6.9 Appendices

6.9.1 Annotation Process

6.9.1.1 Annotation Pipeline

We used the PolitiFact API to retrieve the articles, along with the source pages used in the article, the claim and the author of the claim. For each article we performed the annotation process as follows:

1. Reading the claim and the article.
2. Picking the evidence sentences from the corresponding PolitiFact article. These sentences should sum up the whole article while providing as much evidence as possible.
3. Deciding on the veracity label.
4. Going to each relevant url and checking whether it contains the equivalent textual evidence.

In Step 2, we retrieved evidence sentences sentences, where each sentence follows from the previous one and together they constitute enough evidence to verify the claim and provide an explanation for it.

In Step 4, we wanted to examine how often the evidence can be retrieved from external sources, i.e. not relying on PolitiFact articles. However, we have not gathered enough data to carry out a reliable evaluation of this and thus left the idea for future work.

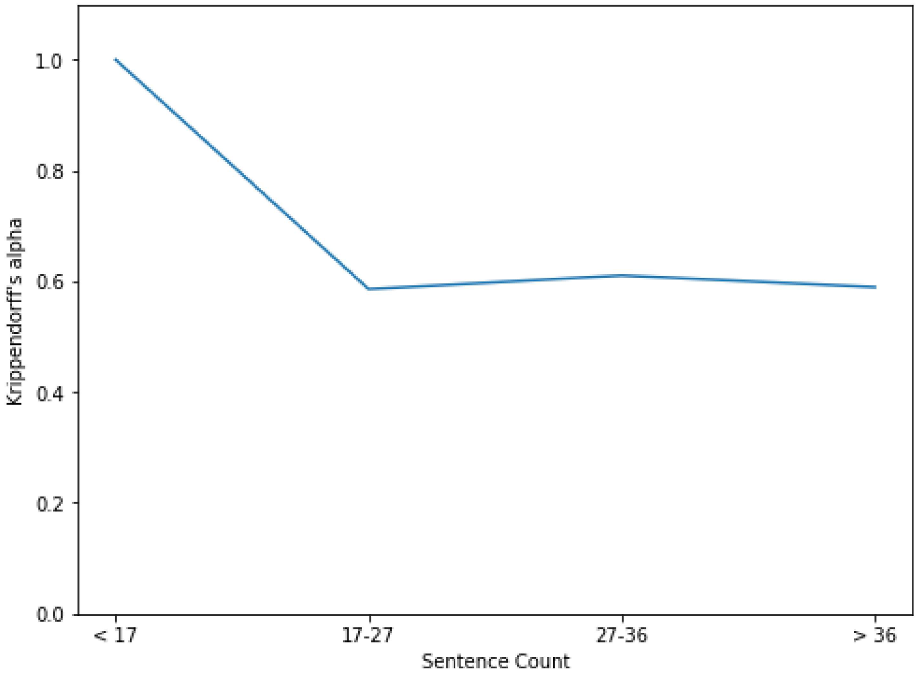


Figure 6.2: Article length vs. inter-annotator agreement.

Originally, following (Thorne et al., 2018), we wanted to have a 'not enough evidence' label, but due to a small frequency of this label in the annotations, as well as due to a significant disagreement between annotators on that label, we decided to discard it and re-label it with one of the remaining labels (false, half-true or true). In case of conflicting label annotations, a third annotator was asked to resolve the conflict.

6.9.1.2 Inter-Annotator Agreement

We report Inter-Annotator Agreement (IAA) agreement on the test set, where we had two annotator annotating each instance. For the veracity prediction task, annotators' Krippendorf's α and Fleiss' κ are equal to 0.638 and 0.637 respectively. By comparison, Thorne et al. (2018) reported Fleiss' κ of 0.684 on the veracity label prediction, which is the another indication of the increased complexity when predicting veracity of claims occurring naturally. For the sentence prediction task, when treating each ruling article as a separate dataset and averaging over all articles, annotators achieve 0.437 Fleiss' κ and 0.437 Krippendorff's α (we also compute IAA when treating all sentences from all articles as one dataset, where both IAA measures drop to 0.400). Figure 6.2 confirms the intuition that annotators tend to agree more on the shorter articles, which are easier to annotate as they contain fewer sets and fewer hops per set.

6.9.2 Training Details

We used LIAR-PLUS and PolitiHop for training in three different settings:

1. Training on LIAR-PLUS only.
2. Training on PolitiHop only.
3. Pre-training on LIAR-PLUS and fine-tuning on PolitiHop.
4. Pre-training on FEVER, then fine-tuning on LIAR-PLUS and PolitiHop.

In the first setting, the models are trained for 4 epochs on LIAR-PLUS. In the second setting, the models are trained for 8 epochs on PolitiHop. In the third setting, models are trained for 4 epochs on LIAR-PLUS, followed by 4 epochs on PolitiHop. In every setting, models are evaluated on the dev set and the model with the best label prediction macro F_1 score is saved, which enables early stopping. For the fourth setting, we pre-train the model for 2 epochs on the FEVER dataset, followed by 4 epochs on LIAR-PLUS, the fine-tune on PolitiHop for 4 epochs.

The models have been trained and evaluated using one NVIDIA TITAN RTX. We report the results based on a single run with a random seed fixed to 42.

Both BERT and Transformer-XH are trained with the same hyperparameters as in (Zhao et al., 2020): BERT 12 layers' with the hidden size of 768, 3 GAT layers with the hidden size of 64. Optimized with Adam with the learning rate of 1e-5. For the TF-IDF baseline, we also remove English stop words using the built-in list in the Scikit-learn library (Pedregosa et al., 2011).

Some of the experiments are ablation studies of the number of GAT layers and the number of retrieved evidence sentences. The former varies between 1 and 7 while the latter varies between 1 and 10.

6.9.3 Additional Results

Number of evidence sentences vs. evidence retrieval performance. One of the challenges of generating fact checking explanations is deciding on the length of the explanation. By design, the explanations should be short, ideally just a few sentences. On the other hand, they have to provide a comprehensive motivation of the fact checking verdict. Transformer-XH handles evidence retrieval by ranking the importance of each input sentence. We, therefore, pick the most highly ranked sentences, according to the model. By default for all experiments in Section 6.5, the top 6 sentences are used, as it is the average length of an annotation in the PolitiHop test set.

Table 6.8 shows how recall trades off against precision and improves as an increasing number of sentences is selected. Test set F_1 grows as the number of sentences grows, while the best train set F_1 is the highest for 3 sentences and it gets worse as the

	Test			Dev		
#S.	**F$_1$**	**Recall**	**Precision**	**F$_1$**	**Recall**	**Precision**
1	15.9	9.7	**62.0**	17.3	13.5	30.5
2	27.6	19.4	61.0	27.8	28.9	**31.2**
3	36.2	28.5	61.2	31.8	39.4	30.0
4	41.3	35.5	59.1	32.3	47.0	27.1
5	44.0	41.0	55.8	31.6	52.9	24.5
6	47.2	47.2	54.5	**32.4**	60.9	23.8
7	49.2	52.5	52.9	31.8	65.8	22.4
8	50.3	56.9	51.0	31.2	70.5	21.2
9	50.8	60.7	49.0	30.5	74.7	20.2
10	**51.0**	**64.1**	47.3	29.1	**76.4**	18.9

Table 6.8: PolitiHop evidence retrieval results for a model trained on LIAR-PLUS full, then fine-tuned on PolitiHop full, with a varying number of top sentences retrieved as evidence. Best number of sentences emboldened.

	Dev			Test		
	Lab	**Evi**	**Joint**	**Lab**	**Evi**	**Joint**
BERT	72.2	43.7	11.3	72.0	43.5	**14.5**
TXH	**73.7**	**44.4**	**12.1**	**72.6**	**44.6**	13.4

Table 6.9: LIAR-PLUS results when trained on the LIAR-PLUS full articles dataset. Best model emboldened. The Lab(el) and Evi(dence) results are F_1 scores, and Joint is measured with FEVER score.

number of sentences increases. F_1 on dev set is much more even for different numbers, but it peaks at 6 sentences. Generally, 6 sentences gives the best trade-off between the performance on test and dev sets, while being short enough to be considered as short summary of the whole article.

LIAR-PLUS. Here, we investigate training then testing on LIAR-PLUS. As Table 6.9 shows, Transformer-XH outperforms BERT by a small margin. This confirms the results in (Zhao et al., 2020) that Transformer-XH generally performs well in multi-hop settings.

Loss function comparison. In this experiment we compare BERT and Transformer-XH performance on the PolitiHop full article setting when trained with three different loss functions. The default loss function is the sum of evidence prediction loss and label prediction loss. The EVI setting uses evidence loss only and saves the model with the highest validation set evidence F_1 score. The LAB setting uses label prediction loss only and saves the model with the highest macro F_1 score on validation data label prediction, just like in the default setting.

	Dev					Test				
	Label		Evidence		Joint	Label		Evidence		Joint
	F$_1$	Acc	F$_1$	Prec	FEVER	F$_1$	Acc	F$_1$	Prec	FEVER
BERT	64.4	75.9	29.6	21.7	28.4	57.8	79.5	45.1	52.2	23.5
Transformer-XH	**64.6**	**78.7**	32.4	23.8	**38.3**	57.3	80.5	**47.2**	**54.5**	**24.5**
BERT-EVI	34.0	51.1	31.8	23.4	26.2	40.7	63.5	45.7	52.8	21.5
Trans-XH-EVI	56.0	68.1	**34.4**	**25.2**	33.3	59.9	75.5	46.7	54.2	21.5
BERT-LAB	59.4	72.3	19.9	14.8	14.9	60.3	77.5	33.8	40.1	13.5
Trans-XH-LAB	62.0	75.2	18.2	13.5	14.9	**60.6**	**81.5**	32.4	38.8	13.0
Random	24.2	27.7	14.7	12.2	0.7	34.1	38.5	22.9	30.2	4.5

Table 6.10: PolitiHop results trained on LIAR-PLUS + PolitiHop full articles datasets. EVI means the model was trained with loss on the evidence prediction task only. LAB means the loss on the label prediction only. The default loss was the sum of both. Best model emboldened.

Table 6.10 shows the best performance with the joint loss for BERT. EVI setting hurts the label prediction while LAB setting hurts the evidence prediction, without providing a clear boost in the second metric over the joint model. Transformer-XH performs much worse on evidence prediction when trained using label prediction loss only. Interestingly, there is no clear performance difference between EVI and default settings.

Adding sentence IDs to sentence encodings. The main goal of this experiment was to see whether providing the information about the positions of the sentences in articles can be leveraged to improve the performance of BERT and Transformer-XH models.

We took the models pre-trained on LIAR-PLUS without sentence positions and fine-tuned the model on PolitiHop with sentence positions, by prepending each sentence's encoding with the token [unusedN], where N =sentence position. Table 6.11 shows a significant performance boost for Transformer-XH label prediction, but not for evidence retrieval. BERT does not exhibit any improvement, which is to be expected as it considers each sentence in isolation, it doesn't learn any interactions between sentences.

Label confidence vs. performance. The goal here was to measure how confident the models are in their label predictions and to see if higher confidence means better performance. The results are presented in Table 6.12.

Transformer-XH is usually sure of its predictions, so not much can be observed based on that - it does indeed have higher F_1 score when it is more confident, but there are too few instances where it is not confident to make any conclusions - apart from the

	Dev					**Test**				
	Label		**Evidence**		**Joint**	**Label**		**Evidence**		**Joint**
	F_1	**Acc**	**F_1**	**Prec**	**FEVER**	**F_1**	**Acc**	**F_1**	**Prec**	**FEVER**
BERT	62.4	75.2	27.7	20.3	24.1	57.4	79.0	42.9	49.2	21.5
Transformer-XH	**65.2**	**78.0**	**32.2**	**23.6**	**38.3**	**66.5**	**84.0**	**47.2**	**54.9**	**26.5**

Table 6.11: `PolitiHop` results for training on LIAR-PLUS `full`, then fine-tuning on `PolitiHop` `full` with sentence ID encodings. Best model emboldened.

	Transformer-XH					**BERT**				
	Label		**Evidence**		**Joint**	**Label**		**Evidence**		**Joint**
	F_1	**Acc**	**F_1**	**Prec**	**FEVER**	**F_1**	**Acc**	**F_1**	**Prec**	**FEVER**
$< 90\%$	**45.5**	**46.2**	**51.6**	**39.7**	**34.6**	30.9	32.6	52.9	41.6	20.9
$\geq 90\%$	67.2	78.3	**54.1**	**40.7**	67.0	**72.5**	**85.7**	50.9	37.8	**67.3**

Table 6.12: `PolitiHop` adversarial dev set performance vs. label confidence.

one that it's often sure but makes a mistake anyway. Besides, NE overlap was not particularly high for the instances where the model got confused.

The effect is even stronger with BERT to the point where having less than 95% confidence usually results in a bad prediction.

6.9.4 PolitiHop Example

Table 6.13 shows an example from the proposed `PolitiHop` dataset.

Claim: Claim: Says 20 million Chinese converted to Islam after it's proven that the coronavirus doesn't affect Muslims. **Speaker**: Viral image **Label**: false
Ruling Comments: [1] Amid fears about the coronavirus disease , a YouTube video offers a novel way to inoculate yourself: convert to Islam. [2] "20m Chinese gets converted to Islam after it is proven that corona virus did not affect the Muslims," reads the title of a video posted online Feb. 18 (...) **[5] That's because the footage is from at least as far back as May 26, 2019, when it was posted on Facebook with this caption: "Alhamdulillah welcome to our brothers in faith." [6] On Nov. 7, 2019, it was posted on YouTube with this title: "MashaaAllah hundreds converted to Islam in Philippines."** [7] Both posts appeared online before the current outbreak of the new coronavirus, COVID-19, was first reported in Wuhan, China, on Dec. 31, 2019. *[8] But even if the footage followed the outbreak, Muslims are not immune to COVID-19, as the Facebook post claims. [9] After China, Iran has emerged as the second focal point for the spread of COVID-19, the New York Times reported on Feb. 24* . [10] "The Middle East is in many ways the perfect place to spawn a pandemic, experts say, with the constant circulation of both Muslim pilgrims and itinerant workers who might carry the virus." [11] On Feb. 18, Newsweek reported that coronavirus "poses a serious risk to millions of inmates in China's Muslim prison camps."

Table 6.13: An example from the PolitiHop dataset. Each example consists of a claim, a speaker (author of the claim), a veracity label and a PolitiFact article with the annotated evidence sentences. One of the evidence sets is in bold, and the other in italics.

Part IV

Diagnostic Explainability Methods

A Diagnostic Study of Explainability Techniques for Text Classification

7

7.1 Introduction

Understanding the rationales behind models' decisions is becoming a topic of pivotal importance, as both the architectural complexity of machine learning models and the number of their application domains increases. Having greater insight into the models' reasons for making a particular prediction has already proven to be essential for discovering potential flaws or biases in medical diagnosis (Caruana et al., 2015) and judicial sentencing (Rich, 2016). In addition, European law has mandated "the right . . . to obtain an explanation of the decision reached" (Regulation, 2016).

Explainability methods attempt to reveal the reasons behind a model's prediction for a single data point, as shown in Figure 7.1. They can be produced post-hoc, i.e., with already trained models. Such post-hoc explanation techniques can be applicable to one specific model (Martens et al., 2008; Wagner et al., 2019) or to a broader range thereof (Ribeiro et al., 2016a; Lundberg and Lee, 2017). They can further be categorised as: employing model gradients (Sundararajan et al., 2017; Simonyan et al., 2013), being perturbation based (Shapley, 1953; Zeiler and Fergus, 2014) or providing explanations through model simplifications (Ribeiro et al., 2016a; Johansson et al., 2004). There also exist explainability methods that generate textual explanations (Camburu et al., 2018) and are trained post-hoc or jointly with the model at hand.

While there is a growing amount of explainability methods, we find that they can produce varying, sometimes contradicting explanations, as illustrated in Figure 7.1. Hence, it is important to *assess existing techniques* and to *provide a generally applicable and automated methodology* for choosing one that is suitable for a particular model architecture and application task (Jacovi and Goldberg, 2020). Robnik-Šikonja and Bohanec (2018) compiles a list of property definitions for explainability techniques, but it remains a challenge to evaluate them in practice. Several other studies have independently proposed different setups for probing varied aspects of explainability techniques (DeYoung et al., 2020a; Sundararajan et al., 2017). However, existing studies evaluating explainability methods are discordant and do not compare to

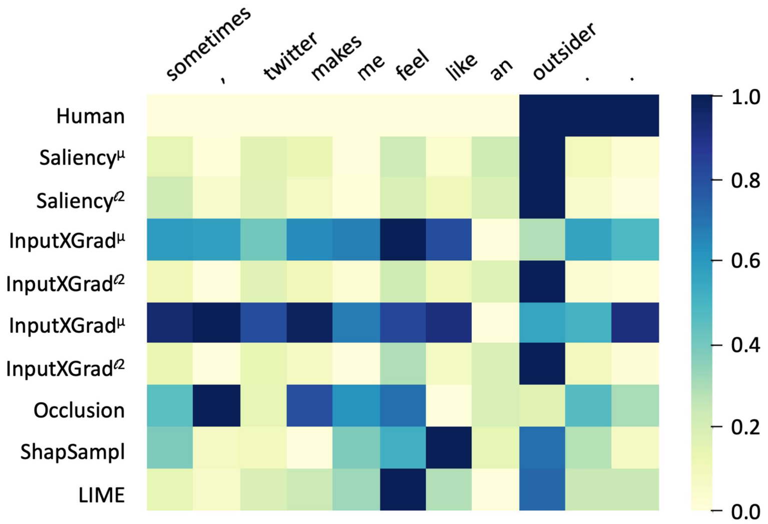


Figure 7.1: Example of the saliency scores for the words (columns) of an instance from the Twitter Sentiment Extraction dataset. They are produced by the explainability techniques (rows) given a `Transformer` model. The first row is the human annotation of the salient words. The scores are normalized in the range $[0, 1]$.

properties from previous studies. In our work, we consider properties from related work and extend them to be applicable to a broader range of downstream tasks.

Furthermore, to create a thorough setup for evaluating explainability methods, one should include at least: (i) different groups of explainability methods (explanation by simplification, gradient-based, etc.), (ii) different downstream tasks, and (iii) different model architectures. However, existing studies usually consider at most two of these aspects, thus providing insights tied to a specific setup.

We propose a number of diagnostic properties for explainability methods and evaluate them in a comparative study. We consider explainability methods from different groups, all widely applicable to most ML models and application tasks. We conduct an evaluation on three text classification tasks, which contain human annotations of salient tokens. Such annotations are available for Natural Language Processing (NLP) tasks, as they are relatively easy to obtain. This is in contrast to ML sub-fields such as image analysis, for which we only found one relevant dataset – 536 manually annotated object bounding boxes for Visual Question Answering (Subramanian et al., 2020).

We further compare explainability methods across three of the most widely used model architectures – `CNN`, `LSTM`, and `Transformer`. The `Transformer` model achieves state-of-the-art performance on many text classification tasks but has a complex architecture, hence methods to explain its predictions are strongly desirable. The proposed properties can also be directly applied to Machine Learning (ML) subfields other than NLP. The code for the paper is publicly available.[1]

In summary, the **contributions** of this work are:

- We compile a comprehensive list of diagnostic properties for explainability and automatic measurement of them, allowing for their effective assessment in practice.
- We study and compare the characteristics of different groups of explainability techniques in three different application tasks and three different model architectures.
- We study the attributions of the explainability techniques and human annotations of salient regions to compare and contrast the rationales of humans and machine learning models.

7.2 Related Work

Explainability methods can be divided into explanations by simplification, e.g., LIME (Ribeiro et al., 2016a); gradient-based explanations (Sundararajan et al., 2017); perturbation-based explanations (Shapley, 1953; Zeiler and Fergus, 2014). Some studies propose the generation of text serving as an explanation, e.g., (Camburu et al., 2018; Lei et al., 2016; Atanasova et al., 2020b). For extensive overviews of existing explainability approaches, see Arrieta et al. (2020).

Explainability methods provide explanations of different qualities, so assessing them systematically is pivotal. A common attempt to reveal shortcomings in explainability techniques is to reveal a model's reasoning process with counter-examples (Alvarez-Melis and Jaakkola, 2018; Kindermans et al., 2019; Atanasova et al., 2020c), finding different explanations for the same output. However, single counter-examples do not provide a measure to evaluate explainability techniques (Jacovi and Goldberg, 2020).

Another group of studies performs human evaluation of the outputs of explainability methods (Lertvittayakumjorn and Toni, 2019; Narayanan et al., 2018). Such studies exhibit low inter-annotator agreement and reflect mostly what appears to be reasonable and appealing to the annotators, not the actual properties of the method.

The most related studies to our work design measures and properties of explainability techniques. Robnik-Šikonja and Bohanec (2018) propose an extensive list of properties. The *Consistency* property captures the difference between explanations of different models that produce the same prediction; and the *Stability* property measures the difference between the explanations of similar instances given a single

[1] https://github.com/copenlu/xai-benchmark

model. We note that similar predictions can still stem from different reasoning paths. Instead, we propose to explore instance activations, which reveal more of the model's reasoning process than just the final prediction. The authors propose other properties as well, which we find challenging to apply in practice. We construct a comprehensive list of diagnostic properties tied with measures that assess the degree of each characteristic.

Another common approach to evaluate explainability methods is to measure the sufficiency of the most salient tokens for predicting the target label (DeYoung et al., 2020a). We also include a sufficiency estimate, but instead of fixing a threshold for the tokens to be removed, we measure the decrease of a model's performance, varying the proportion of excluded tokens. Other perturbation-based evaluation studies and measures exist (Sundararajan et al., 2017; Adebayo et al., 2018), but we consider the above, as it is the most widely applied.

Another direction of explainability evaluation is to compare the agreement of salient words annotated by humans to the saliency scores assigned by explanation techniques (DeYoung et al., 2020a). We also consider the latter and further study the agreement across model architectures, downstream tasks, and explainability methods. While we consider human annotations at the word level (Camburu et al., 2018; Lei et al., 2016), there are also datasets (Clark et al., 2019; Khashabi et al., 2018) with annotations at the sentence-level, which would require other model architectures, so we leave this for future work.

Existing studies for evaluating explainability heavily differ in their scope. Some concentrate on a **single model architecture** - BERT-LSTM (DeYoung et al., 2020a), RNN (Arras et al., 2019), CNN (Lertvittayakumjorn and Toni, 2019), whereas a few consider **more than one** model (Guan et al., 2019; Poerner et al., 2018). Some studies concentrate on one **particular dataset** (Guan et al., 2019; Arras et al., 2019), while only a few generalize their findings over **downstream tasks** (DeYoung et al., 2020a; Vashishth et al., 2019). Finally, existing studies focus on one (Vashishth et al., 2019) or a single group of explainability methods (DeYoung et al., 2020a; Adebayo et al., 2018). Our study is the first to propose a unified comparison of different groups of explainability techniques across three text classification tasks and three model architectures.

7.3 Evaluating Attribution Maps

We now define a set of diagnostic properties of explainability techniques, and propose how to quantify them. Similar notions can be found in related work (Robnik-Šikonja and Bohanec, 2018; DeYoung et al., 2020a), and we extend them to be generally applicable to downstream tasks. We first introduce the prerequisite notation. Let $X = \{(x_i, y_i, w_i) | i \in [1, N]\}$ be the test dataset, where each instance consists of a list of *tokens* $x_i = \{x_{i,j} | j \in [1, |x_i|]\}$, a *gold label* y_i, and a *gold saliency score* for each of the tokens in x_i: $w_i = \{w_{i,j} | j \in [1, |x_i|]\}$ with each $w_{i,j} \in \{0, 1\}$. Let ω be an explanation technique that, given a model M, a class c, and a single instance x_i, computes saliency scores for each token in the input: $\omega^M_{x_i,c}= \{\omega^M_{(i,j),c} | j \in [1, |x_i|]\}$. Finally, let $M = M_1, \ldots M_K$ be models with the same architecture, each trained from a randomly chosen seed, and let $M' = M'_1, \ldots M'_K$ be models of the same architecture, but with randomly initialized weights.

Agreement with human rationales (HA). This diagnostic property measures the degree of overlap between saliency scores provided by human annotators, specific to the particular task, and the word saliency scores computed by an explainability technique on each instance. The property is a simple way of approximating the quality of the produced feature attributions. While it does not necessarily mean that the saliency scores explain the predictions of a model, we assume that explanations with high agreement scores would be more comprehensible for the end-user as they would adhere more to human reasoning. With this diagnostic property, we can also compare how the type and the performance of a model and/or dataset affect the agreement with human rationales when observing one type of explainability technique.

During evaluation, we provide an estimate of the average agreement of the explainability technique across the dataset. To this end, we start at the instance level and compute the Average Precision (AP) of produced saliency scores $\omega^M_{x_i,c}$ by comparing them to the gold saliency annotations w_i. Here, the label for computing the saliency scores is the gold label: $c = y_i$. Then, we compute the average across all instances, arriving at Mean AP (MAP):

$$\text{MAP}(\omega, M, X) = \frac{1}{N} \sum_{i \in [1,N]} AP(w_i, \omega^M_{x_i,y_i}) \tag{7.1}$$

Confidence Indication (CI). A token from a single instance can receive several saliency scores, indicating its contribution to the prediction of each of the classes. Thus, when a model recognizes a highly indicative pattern of the predicted class k, the tokens involved in the pattern would have highly positive saliency scores for this class and highly negative saliency scores for the remaining classes. On the other hand, when the model is not highly confident, we can assume that it is unable to recognize a strong indication of any class, and the tokens accordingly do not have high saliency scores for any class. Thus, the computed explanation of an instance i should indicate the confidence $p_{i,k}$ of the model in its prediction.

We propose to measure the predictive power of the produced explanations for the confidence of the model. We start by computing the Saliency Distance (SD) between the saliency scores for the predicted class k to the saliency scores of the other classes K/k (Eq. 7.2). Given the distance between the saliency scores, we predict the confidence of the class with logistic regression (LR) and finally compute the Mean Absolute Error – MAE (Eq. 7.3), of the predicted confidence to the actual one.

$$\text{SD} = \sum_{j \in [0,|x|]} D(\omega^M_{x_{i,j},k}, \omega^M_{x_{i,j},K/k}) \tag{7.2}$$

$$\text{MAE}(\omega, M, X) = \sum_{i \in [1,N]} |p_{i,k} - \text{LR}(\text{SD})| \tag{7.3}$$

For tasks with two classes, D is the subtraction of the saliency value for class k and the other class. For more than two classes, D is the concatenation of the max, min, and average across the differences of the saliency value for class k and the other classes. Low MAE indicates that model's confidence can be easily identified by looking at the produced explanations.

Faithfulness (F). Since explanation techniques are employed to explain model predictions for a single instance, an essential property is that they are faithful to the model's inner workings and not based on arbitrary choices. A well-established way of measuring this property is by replacing a number of the most-salient words with a mask token (DeYoung et al., 2020a) and observing the drop in the model's performance. To avoid choosing an unjustified percentage of words to be perturbed, we produce several dataset perturbations by masking 0, 10, 20, ..., 100% of the tokens in order of decreasing saliency, thus arriving at X^{ω^0}, $X^{\omega^{10}}$, ..., $X^{\omega^{100}}$. Finally, to produce a single number to measure faithfulness, we compute the area under the threshold-performance curve (AUC-TP):

$$\begin{gathered}\text{AUC-TP}(\omega, M, X) = \\ \text{AUC}([(i, P(M(X^{\omega^0})) - M(X^{\omega^i}))])\end{gathered} \tag{7.4}$$

where P is a task specific performance measure and $i \in [0, 10, \ldots, 100]$. We also compare the AUC-TP of the saliency methods to a random saliency map to find whether there are explanation techniques producing saliency scores without any contribution over a random score.

Using AUC-TP, we perform an ablation analysis which is a good approximation of whether the most salient words are also the most important ones for a model's prediction. However, some prior studies (Feng et al., 2018) find that models remain confident about their prediction even after stripping most input tokens, leaving a few that might appear nonsensical to humans. The diagnostic properties that follow aim to facilitate a more in-depth analysis of the alignment between the inner workings of a model and produced saliency maps.

Rationale Consistency (RC). A desirable property of an explainability technique is to be consistent with the similarities in the reasoning paths of several models on a single instance. Thus, when two reasoning paths are similar, the scores provided by an explainability technique ω should also be similar, and vice versa. Note that we are interested in similar reasoning paths as opposed to similar predictions, as the latter does not guarantee analogous model rationales. For models with diverse architectures, we expect rationales to be diverse as well and to cause low consistency. Therefore, we focus on a set of models with the same architecture, trained from different random seeds as well as the same architecture, but with randomly initialized weights. The latter would ensure that we can have model pairs (M_s, M_p) with similar and distant rationales. We further claim that the similarity in the reasoning paths could be measured effectively with the distance between the activation maps (averaged across layers and neural nodes) produced by two distinct models (Eq. 7.5). The distance between the explanation scores is computed simply by subtracting the two (Eq. 7.6). Finally, we compute Spearman's ρ between the similarity of the explanation scores and the similarity of the attribution maps (Eq. 7.7).

$$D(M_s, M_p, x_i) = D(M_s(x_i), M_p(x_i)) \tag{7.5}$$

$$D(M_s, M_p, x_i, \omega) = D(\omega^{M_s}_{x_i,y_i}, \omega^{M_p}_{x_i,y_i}) \tag{7.6}$$

$$\begin{aligned}\rho(M_s, M_p, X, \omega) = \rho(D(M_s, M_p, x_i),\\ D(M_s, M_p, x_i, \omega) | i \in [1, N])\end{aligned} \tag{7.7}$$

The higher the positive correlation is, the more consistent the attribution method would be. We choose Spearman's ρ as it measures the monotonic correlation between the two variables. On the other hand, Pearson's ρ measures only the linear correlation, and we can have a non-linear correlation between the activation difference and the saliency score differences. When subtracting saliency scores and layer activations, we also take the absolute value of the vector difference as the property should be invariant to order of subtraction. An additional benefit of the property is that low correlation scores would also help to identify explainability techniques that are not faithful to a model's rationales.

Dataset Consistency (DC). The next diagnostic property is similar to the above notion of rationale consistency but focuses on consistency across instances of a dataset as opposed to consistency across different models of the same architecture. In this case,

Dataset	Example	Size	Length
e-SNLI (Camburu et al., 2018)	*Premise:* An adult dressed in black **holds a stick.** *Hypothesis:* An adult is walking away, **empty-handed**. *Label*: contradiction	549 367 Train 9 842 Dev 9 824 Test	27.4 inst. 5.3 expl.
Movie Reviews (Zaidan et al., 2007)	*Review:* he is one of **the most exciting martial artists on the big screen**, continuing to perform his own stunts and **dazzling audiences** with his flashy kicks and punches. *Class:* Positive	1 399 Train 199 Dev 199 Test	834.9 inst. 56.18 expl.
Tweet Sentiment Extraction (TSE) [2]	*Tweet:* im soo **bored**...im deffo missing my music channels *Class:* Negative	21 983 Train 2 747 Dev 2 748 Test	20.5 inst. 9.99 expl.

Table 7.1: Datasets with human-annotated saliency explanations. The *Size* column presents the dataset split sizes we use in our experiments. The *Length* column presents the average number of instance tokens in the test set *(inst.)* and the average number of human annotated explanation tokens *(expl.)*.

we test whether instances with similar rationales also receive similar explanations. While Rationale Consistency compares instance explanations of the same instance for different model rationales, Dataset Consistency compares explanations for pairs of instances on the same model. We again measure the similarity between instances x_i and x_j by comparing their activation maps, as in Eq. 7.8. The next step is to measure the similarity of the explanations produced by an explainability technique ω, which is the difference between the saliency scores as in Eq. 7.9. Finally, we measure Spearman's ρ between the similarity in the activations and the saliency scores as in Eq. 7.10. We again take the absolute value of the difference.

$$D(M, x_i, x_j) = D(M(x_i), M(x_j)) \tag{7.8}$$

$$D(M, x_i, x_j, \omega) = D(\omega^M_{x_i,y_i}, \omega^M_{x_j,y_i}) \tag{7.9}$$

$$\begin{aligned} \rho(M, X, \omega) = \rho(D(M, x_i, x_j), \\ D(M, x_i, x_j, \omega) | i, j \in [1, N]) \end{aligned} \tag{7.10}$$

7.4 Experiments

7.4.1 Datasets

Table 7.1 provides an overview of the used datasets. For e-SNLI, models predict inference – contradiction, neutral, or entailment – between sentence tuples. For the Movie Reviews dataset, models predict the sentiment – positive, negative, or neutral – of reviews with multiple sentences. Finally, for the TSE dataset, models predict tweets' sentiment – positive, negative, or neutral. The e-SNLI dataset provides three dataset splits with human-annotated rationales, which we use as training, dev, and test sets, respectively. The Movie Reviews dataset provides rationale annotations for nine out of ten splits. Hence, we use the ninth split as a test and the eighth split as a dev set, while the rest are used for training. Finally, the TSE dataset only provides rationale annotations for the training dataset, and we therefore randomly split it into 80/10/10% chunks for training, development and testing.

7.4.2 Models

We experiment with different commonly used base models, namely `CNN` (Fukushima, 1980), `LSTM` (Hochreiter and Schmidhuber, 1997), and the `Transformer` (Vaswani et al., 2017) architecture BERT (Devlin et al., 2019). The selected models allow for a comparison of the explainability techniques on diverse model architectures. Table 7.2 presents the performance of the separate models on the datasets.

For the `CNN` model, we use an embedding, a convolutional, a max-pooling, and a linear layer. The embedding layer is initialized with GloVe (Pennington et al., 2014) embeddings and is followed by a dropout layer. The convolutional layer computes convolutions with several window sizes and multiple-output channels with ReLU (Hahnloser et al., 2000) as an activation function. The result is compressed down with a max-pooling layer, passed through a dropout layer, and into a fine linear layer responsible for the prediction. The final layer has a size equal to the number of classes in the dataset.

The `LSTM` model again contains an embedding layer initialized with the GloVe embeddings. The embeddings are passed through several bidirectional LSTM layers. The final output of the recurrent layers is passed through three linear layers and a final dropout layer.

For the `Transformer` model, we fine-tune the pre-trained basic, uncased language model (LM) (Wolf et al., 2019). The fine-tuning is performed with a linear layer on top of the LM with a size equal to the number of classes in the corresponding task. Further implementation details for all of the models, as well as their F_1 scores, are presented in 7.7.1.

[2] https://www.kaggle.com/c/tweet-sentiment-extraction

Model	Val	Test
e-SNLI		
Transformer	0.897 (±0.002)	0.892 (±0.002)
Transformer^{RI}	0.167 (±0.003)	0.167 (±0.003)
CNN	0.773 (±0.003)	0.768 (±0.002)
CNN^{RI}	0.195 (±0.038)	0.194 (±0.037)
LSTM	0.794 (±0.005)	0.793 (±0.009)
LSTM^{RI}	0.176 (±0.013)	0.176 (±0.000)
Movie Reviews		
Transformer	0.859 (±0.044)	0.856 (±0.018)
Transformer^{RI}	0.335 (±0.003)	0.333 (±0.000)
CNN	0.831 (±0.014)	0.773 (±0.005)
CNN^{RI}	0.343 (±0.020)	0.333 (±0.001)
LSTM	0.614 (±0.017)	0.567 (±0.019)
LSTM^{RI}	0.362 (±0.030)	0.363 (±0.041)
TSE		
Transformer	0.772 (±0.005)	0.781 (±0.009)
Transformer^{RI}	0.165 (±0.025)	0.171 (±0.022)
CNN	0.708 (±0.007)	0.730 (±0.007)
CNN^{RI}	0.221 (±0.060)	0.226 (±0.055)
LSTM	0.701 (±0.005)	0.727 (±0.004)
LSTM^{RI}	0.196 (±0.070)	0.204 (±0.070)

Table 7.2: Models' F_1 score on the test and the validation datasets. The results present the average and the standard deviation of the Performance measure over five models trained from different seeds. The random versions of the models are again five models, but only randomly initialized, without training.

7.4.3 Explainability Techniques

We select the explainability techniques to be representative of different groups – gradient (Sundararajan et al., 2017; Simonyan et al., 2013), perturbation (Shapley, 1953; Zeiler and Fergus, 2014) and simplification based (Ribeiro et al., 2016a; Johansson et al., 2004).

Starting with the **gradient-based** approaches, we select *Saliency* (Simonyan et al., 2013) as many other gradient-based explainability methods build on it. It computes the gradient of the output w.r.t. the input. We also select two widely used improvements of the *Saliency* technique, namely *InputXGradient* (Kindermans et al., 2016), and *Guided Backpropagation* (Springenberg et al., 2014). InputXGradient additionally multiplies the gradient with the input and *Guided Backpropagation* overwrites the gradients of ReLU functions so that only non-negative gradients are backpropagated.

From the **perturbation-based** approaches, we employ *Occlusion* (Zeiler and Fergus, 2014), which replaces each token with a baseline token (as per standard, we use the value zero) and measures the change in the output. Another popular perturbation-based technique is the *Shapley Value Sampling* (Castro et al., 2009). It is based on the Shapley Values approach that computes the average marginal contribution of each word across all possible word perturbations. The Sampling variant allows for a faster approximation of Shapley Values by considering only a fixed number of random perturbations as opposed to all possible perturbations.

Finally, we select the **simplification-based** explanation technique LIME (Ribeiro et al., 2016a). For each instance in the dataset, LIME trains a linear model to approximate the local decision boundary for that instance.

Generating explanations. The saliency scores from each of the explainability methods are generated for each of the classes in the dataset. As all of the gradient approaches provide saliency scores for the embedding layer (the last layer that we can compute the gradient for), we have to aggregate them to arrive at one saliency score per input token. As we found different aggregation approaches in related studies (Bansal et al., 2016; DeYoung et al., 2020a), we employ the two most common methods – L2 norm and averaging (denoted as μ and $\ell 2$ in the explainability method names).

7.5 Results and Discussion

We report the measures of each diagnostic property as well as FLOPs as a measure of the computing time used by the particular method. For all diagnostic properties, we also include the randomly assigned saliency as a baseline.

7.5.1 Results

Of the three model architectures, unsurprisingly, the `Transformer` model performs best, while the `CNN` and the `LSTM` models are close in performance. It is only for the IMDB dataset that the `LSTM` model performs considerably worse than the `CNN`, which we attribute to the fact that the instances contain a large number of tokens, as shown in Table 7.1. As this is not the core focus of this paper, detailed results can be found in the supplementary material.

Overall results. Table 7.3 presents the mean of all properties across tasks and models with all property measures normalized to be in the range [0,1]. We see that gradient-based explainability techniques always have the best or the second-best performance for the diagnostic properties across all three model architectures and all three downstream tasks. Note that, *InputXGrad*$^{\mu}$ and *GuidedBP*$^{\mu}$, which are computed with a mean aggregation of the scores, have some of the worst results. We conjecture that this is due to the large number of values that are averaged – the mean smooths out any differences in the values. In contrast, the L2 norm aggregation amplifies

Model	**Saliency**	**e-SNLI**	**IMDB**	**TSE**
`Transformer`	*Random*	0.201	0.517	0.185
	ShapSampl	0.479	0.481	0.667
	LIME	**0.809**	0.604	0.553
	Occlusion	0.523	0.323	0.556
	Saliency$^{\mu}$	0.772	0.671	0.707
	Saliency$^{\ell 2}$	0.781	**0.687**	0.696
	InputXGrad$^{\mu}$	0.364	0.432	0.307
	InputXGrad$^{\ell 2}$	0.796	0.676	**0.754**
	GuidedBP$^{\mu}$	0.468	0.236	0.287
	GuidedBP$^{\ell 2}$	0.782	0.676	0.685
`CNN`	*Random*	0.209	0.468	0.384
	ShapSampl	0.460	0.648	0.630
	LIME	0.571	0.572	**0.681**
	Occlusion	0.554	0.411	0.594
	Saliency$^{\mu}$	0.853	0.712	0.595
	Saliency$^{\ell 2}$	0.875	**0.796**	0.631
	InputXGrad$^{\mu}$	0.576	0.662	0.613
	InputXGrad$^{\ell 2}$	**0.881**	0.759	0.636
	GuidedBP$^{\mu}$	0.403	0.346	0.438
	GuidedBP$^{\ell 2}$	0.875	0.788	0.628
`LSTM`	*Random*	0.166	0.343	0.225
	ShapSampl	0.606	0.605	0.526
	LIME	0.759	0.233	0.630
	Occlusion	0.609	0.589	0.681
	Saliency$^{\mu}$	0.795	0.568	0.702
	Saliency$^{\ell 2}$	0.800	0.583	**0.704**
	InputXGrad$^{\mu}$	0.432	0.481	0.441
	InputXGrad$^{\ell 2}$	**0.820**	**0.685**	0.693
	GuidedBP$^{\mu}$	0.492	0.553	0.410
	GuidedBP$^{\ell 2}$	0.805	0.660	**0.720**

Table 7.3: Mean of the diagnostic property measures for all tasks and models. The best result for the particular model architecture and downstream task is in bold and the second-best is underlined.

the presence of large and small values in the vector. From the non-gradient based explainability methods, *LIME* has the best performance, where in two out of nine cases it has the best performance. It is followed by *ShapSampl* and *Occlusion*. We can conclude that the occlusion based methods overall have the worst performance according to the diagnostic properties.

Furthermore, we see that the explainability methods achieve better performance for the e-SNLI and the TSE datasets with the `Transformer` and `LSTM` architectures, whereas the results for the IMDB dataset are the worst. We hypothesize that this is due to the longer text of the input instances in the IMDB dataset. The scores also indicate that the explainability techniques have the highest diagnostic property measures for the `CNN` model with the e-SNLI and the IMDB datasets, followed by the `LSTM`, and the `Transformer` model. We suggest that the performance of the explainability tools can

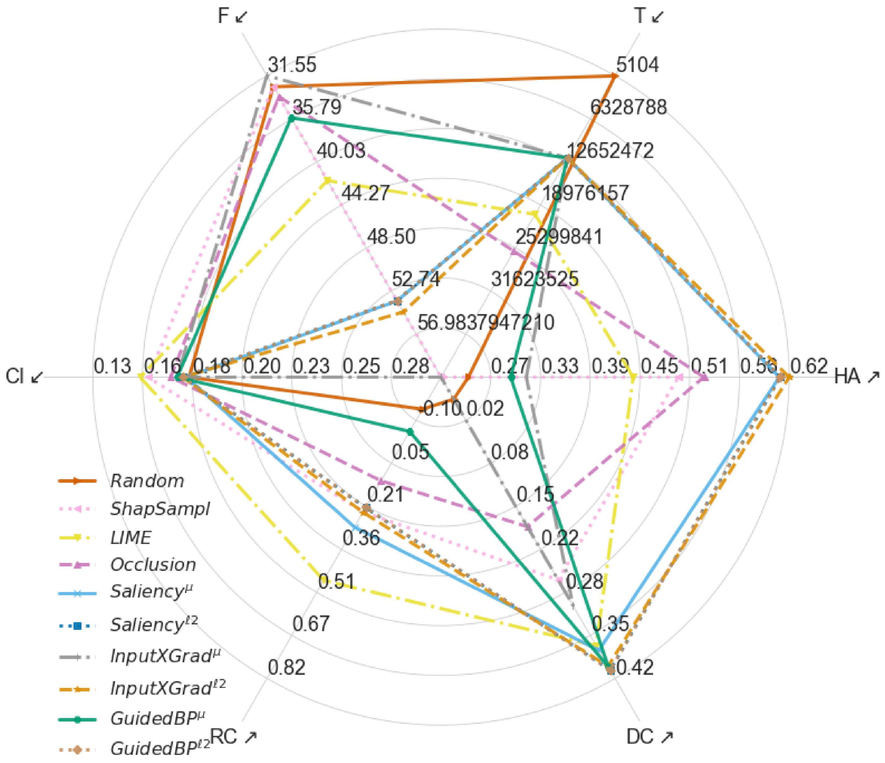


Figure 7.2: Diagnostic property evaluation for all explainability techniques, on the e-SNLI dataset, `Transformer` model. The ↗ and ↙ signs indicate that higher, correpspondingly lower, values of the property measure are better.

be worse for large complex architectures with a huge number of neural nodes, like the `Transformer` one, and perform better for small, linear architectures like the `CNN`.

Diagnostic property performance. Figures 7.2, 7.3, 7.4 show the performance of each explainability technique for all diagnostic properties on the e-SNLI dataset. The TSE and IMDB datasets show similar tendencies and corresponding figures can be found in the supplementary material.

Agreement with human rationales. We observe that the best performing explainability technique for the `Transformer` model is *InputXGrad*$^{\ell2}$ followed by the gradient-based ones with L2 norm aggregation. While for the `CNN` and the `LSTM` models, we observe similar trends, their MAP scores are always lower than for the `Transformer`, which indicates a correlation between the performance of a model and its agreement with human rationales. Furthermore, the MAP scores of the `CNN` model are higher than for the `LSTM` model, even though the latter achieves higher F_1 scores on the e-SNLI dataset. This might indicate that the representations of the `LSTM` model are less in line with human rationales. Finally, we note that the mean aggregations of the gradient-based explainability techniques have MAP scores close to or even worse than those from the randomly initialized models.

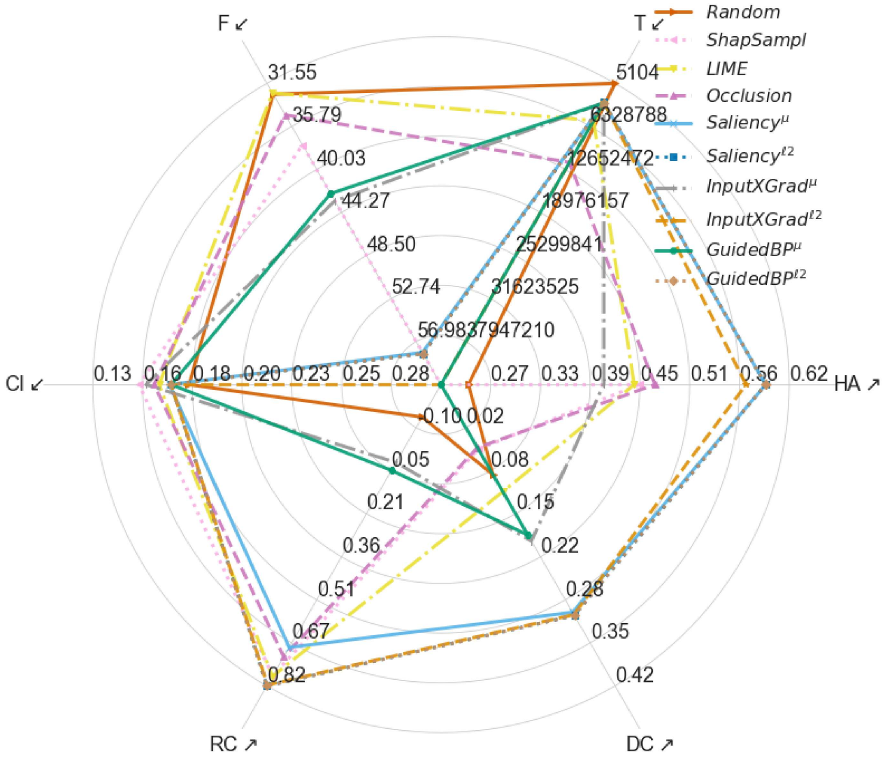


Figure 7.3: Diagnostic property evaluation for all explainability techniques, on the e-SNLI dataset, CNN model. The ↗ and ↙ signs indicate that higher, correpspondingly lower, values of the property measure are better.

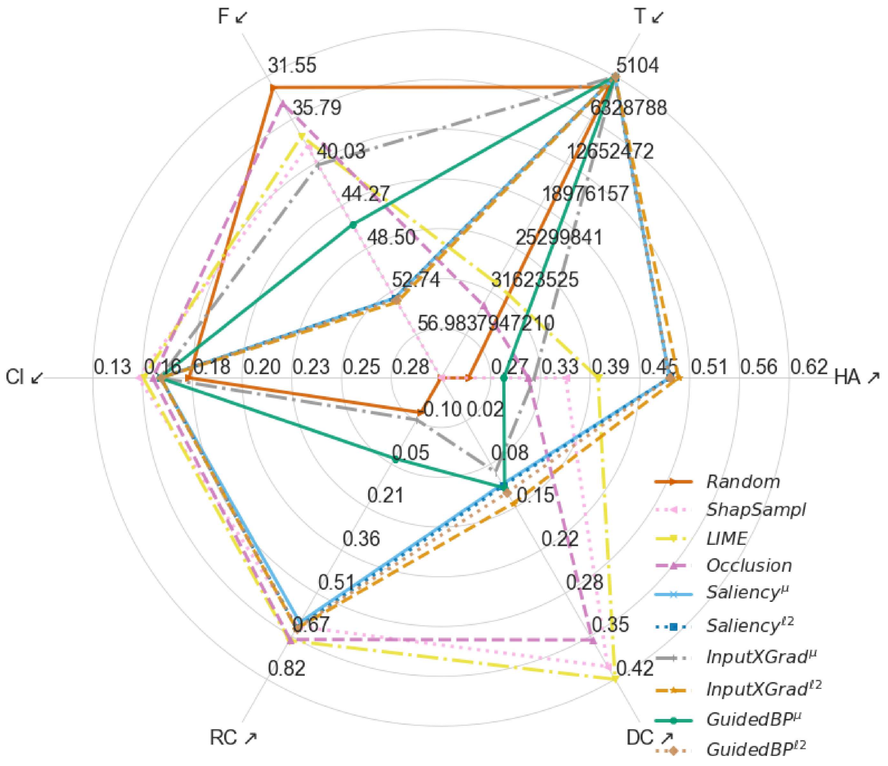


Figure 7.4: Diagnostic property evaluation for all explainability techniques, on the e-SNLI dataset, LSTM model. The ↗ and ↙ signs indicate that higher, correpspondingly lower, values of the property measure are better.

Faithfulness. We find that gradient-based techniques have the best performance for the Faithfulness diagnostic property. On the e-SNLI dataset, it is particularly *InputXGrad*$^{\ell 2}$, which performs well across all model architectures. We further find that the CNN exhibits the highest Faithfulness scores for seven out of nine explainability methods. We hypothesize that this is due to the simple architecture with relatively few neural nodes compared to the recurrent nature of the LSTM model and the large number of neural nodes in the Transformer architecture. Finally, models with high Faithfulness scores do not necessarily have high Human agreement scores and vice versa. This suggests that these two are indeed separate diagnostic properties, and the first should not be confused with estimating the faithfulness of the techniques.

Confidence Indication. We find that the Confidence Indication of all models is predicted most accurately by the *ShapSampl*, *LIME*, and *Occlusion* explainability methods. This result is expected, as they compute the saliency of words based on differences in the model's confidence using different instance perturbations. We further find that the CNN model's confidence is better predicted with *InputXGrad*$^{\mu}$. The lowest MAE with the balanced dataset is for the CNN and LSTM models. We hypothesize that this could be due to these models' overconfidence, which makes it challenging to detect when the model is not confident of its prediction.

Rationale Consistency. There is no single universal explainability technique that achieves the highest score for Rationale Consistency property. We see that *LIME* can be good at achieving a high performance, which is expected, as it is trained to approximate the model's performance. The latter is beneficial, especially for models with complex architectures like the Transformer. The gradient-based approaches also have high Rationale Consistency scores. We find that the *Occlusion* technique is the best performing for the LSTM across all tasks, as it is the simplest of the explored explainability techniques, and does not inspect the model's internals or try to approximate them. This might serve as an indication that LSTM models, due to their recurrent nature, can be best explained with simple perturbation based methods that do not examine a model's reasoning process.

Dataset Consistency. Finally, the results for the Dataset Consistency property show low to moderate correlations of the explainability techniques with similarities across instances in the dataset. The correlation is present for LIME and the gradient-based techniques, again with higher scores for the L2 aggregated gradient-based methods.

Overall. To summarise, the proposed list of diagnostic properties allows for assessing existing explainability techniques from different perspectives and supports the choice of the best performing one. Individual property results indicate that gradient-based methods have the best performance. The only strong exception to the above is the better performance of *ShapSampl* and *LIME* for the Confidence Indication diagnostic property. However, *ShapSampl*, *LIME* and *Occlusion* take considerably more time to compute and have worse performance for all other diagnostic properties.

7.6 Conclusion

We proposed a comprehensive list of diagnostic properties for the evaluation of explainability techniques from different perspectives. We further used them to compare and contrast different groups of explainability techniques on three downstream tasks and three diverse architectures. We found that gradient-based explanations are the best for all of the three models and all of the three downstream text classification tasks that we consider in this work. Other explainability techniques, such as *ShapSampl*, *LIME* and *Occlusion* take more time to compute, and are in addition considerably less faithful to the models and less consistent with the rationales of the models and similarities in the datasets.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

7.7 Appendices

7.7.1 Experimental Setup

Machine Learning Models. The models used in our experiments are trained on the training splits, and the parameters are selected according to the development split. We conducted fine-tuning in a grid-search manner with the ranges and parameters we describe next. We use superscripts to indicate when a parameter value was selected for one of the datasets e-SNLI – 1, Movie Review – 2, and TSE – 3. For the `CNN` model, we experimented with the following parameters: embedding dimension $\in$ {50, 100, 200, $300^{1,2,3}$}, batch size $\in$ {16^{2}, 32, 64^{3}, 128, 256^{1}}, dropout rate $\in$ {$0.05^{1,2,3}$, 0.1, 0.15, 0.2}, learning rate for an Adam optimizer $\in$ {0.01, 0.03, $0.001^{2,3}$, 0.003, 0.0001^{1}, 0.0003}, window sizes $\in$ {$[2, 3, 4]^{2}$, [2, 3, 4, 5], $[3, 4, 5]^{3}$, [3, 4, 5, 6],[4, 5, 6], $[4, 5, 6, 7]^{1}$}, and number of output channels $\in$ {$50^{2,3}$, 100, 200, 300^{1}}. We leave the stride and the padding parameters to their default values – one and zero.

For the `LSTM` model we fine-tuned over the following grid of parameters: embedding dimension $\in$ {50, $100^{1,2}$, 200^{3}, 300}, batch size $\in$ {$16^{2,3}$, 32, 64, 128, 256^{1}}, dropout rate $\in$ {0.05^{3}, $0.1^{1,2}$, 0.15, 0.2}, learning rate for an Adam optimizer $\in$ {0.01^{1}, 0.03^{2},

Model	Time	Score
e-SNLI		
Transformer	244.763 (±62.022)	0.523 (±0.356)
CNN	195.041 (±53.994)	0.756 (±0.028)
LSTM	377.180 (±232.918)	0.708 (±0.205)
Movie Reviews		
Transformer	3.603 (±0.031)	0.785 (±0.226)
CNN	4.777 (±1.953)	0.756 (±0.058)
LSTM	5.344 (±1.593)	0.584 (±0.061)
TSE		
Transformer	9.393 (±1.841)	0.783 (±0.006)
CNN	2.240 (±0.544)	0.730 (±0.035)
LSTM	3.781 (±1.196)	0.713 (±0.076)

Table 7.4: Hyper-parameter tuning details. *Time* is the average time (mean and standard deviation in brackets) measured in minutes required for a particular model with all hyper-parameter combinations. *Score* is the mean and standard deviation of the performance on the validation set as a function of the number of the different hyper-parameter searches.

$0.001^{2,3}$, 0.003, 0.0001, 0.0003}, number of LSTM layers ∈ {$1^{2,3}$, 2, 3, 4^{1}}, LSTM hidden layer size ∈ {50, $100^{1,2,3}$, 200, 300}, and size of the two linear layers ∈ {$[50, 25]^{2}$, $[100, 50]^{1}$, $[200, 100]^{3}$}. We also experimented with other numbers of linear layers after the recurrent ones, but having three of them, where the final was the prediction layer, yielded the best results.

The CNN and LSTM models are trained with an early stopping over the validation accuracy with a patience of five and a maximum number of training epochs of 100. We also experimented with other optimizers, but none yielded improvements.

Finally, for the Transformer model we fine-tuned the pre-trained basic, uncased LM (Wolf et al., 2019)(110M parameters) where the maximum input size is 512, and the hidden size of each layer of the 12 layers is 768. We performed a grid-search over learning rate of $\in \{1e-5, 2e-5^{1,2}, 3e-5^{3}, 4e-5, 5e-5\}$. The models were trained with a warm-up period where the learning rate increases linearly between 0 and 1 for 0.05% of the steps found with a grid-search. We train the models for five epochs with an early stopping with patience of one as the Transformer models are easily fine-tuned for a small number of epochs.

All experiments were run on a single NVIDIA TitanX GPU with 8GB, and 4GB of RAM and 4 Intel Xeon Silver 4110 CPUs.

The models were evaluated with macro F_1 score, which can be found here https://scikit-learn.org/stable/modules/generated/sklearn.metrics and is defined as follows:

$$Precision(P) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$Recall(R) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$F_1 = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

Explainability generation. When evaluating the Confidence Indication property of the explainability measures, we train a logistic regression for 5 splits and provide the MAE over the five test splits. As for some of the models, e.g. `Transformer`, the confidence is always very high, the LR starts to predict only the average confidence. To avoid this, we additionally randomly up-sample the training instances with a smaller confidence, making the number of instances in each confidence interval [0.0-0.1],... [0.9-1.0]) to be the same as the maximum number of instances found in one of the separate intervals.

For both Rationale and Dataset Consistency properties, we consider Spearman's ρ. While Pearson's ρ measures only the linear correlation between two variables (a change in one variable should be proportional to the change in the other variable), Spearman's ρ measures the monotonic correlation (when one variable increases, the other increases, too). In our experiments, we are interested in the monotonic correlation as all activation differences don't have to be linearly proportional to the differences of the explanations and therefore measure Spearman's ρ.

The Dataset Consistency property is estimated over instance pairs from the test dataset. As computing it for all possible pairs in the dataset is computationally expensive, we select 2 000 pairs from each dataset in order of their decreasing word overlap and sample 2 000 from the remaining instance pairs. This ensures that we compute the diagnostic property on a set containing tuples of similar and different instances.

Both the Dataset Consistency property and the Rationale Consistency property estimate the difference between the instances based on their activations. For the `LSTM` model, the activations of the LSTM layers are limited to the output activation also used for prediction as it isn't possible to compare activations with different lengths due to the different token lengths of the different instances. We also use min-max

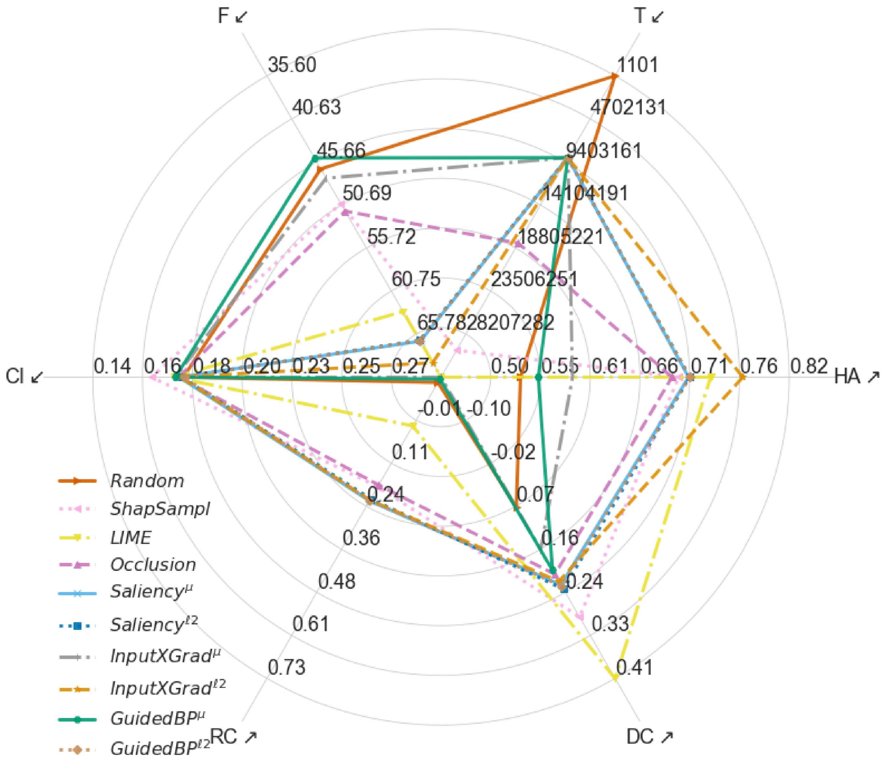


Figure 7.5: Diagnostic property evaluation for all explainability techniques, on the TSE dataset, `Transformer` model. The ↗ and ↙ signs indicate that higher, correspondingly lower, values of the property measure are better.

scaling of the differences in the activations and the saliencies as the saliency scores assigned by some explainability techniques are very small.

7.7.2 Spider Figure for the IMDB dataset

Figures 7.5, 7.6, 7.7 present the diagnostic property evaluation for the TSE dataset for the `Transformer`, `CNN`, `LSTM` models correspondingly. Figures 7.8, 7.9, 7.10 present the diagnostic property evaluation for the IMDB dataset for the `Transformer`, `CNN`, `LSTM` models correspondingly.

7.7.3 Detailed explainability techniques evaluation results.

Tables 7.5, 7.6, 7.7, 7.8, and 7.9 present detailed diagnostic property results.

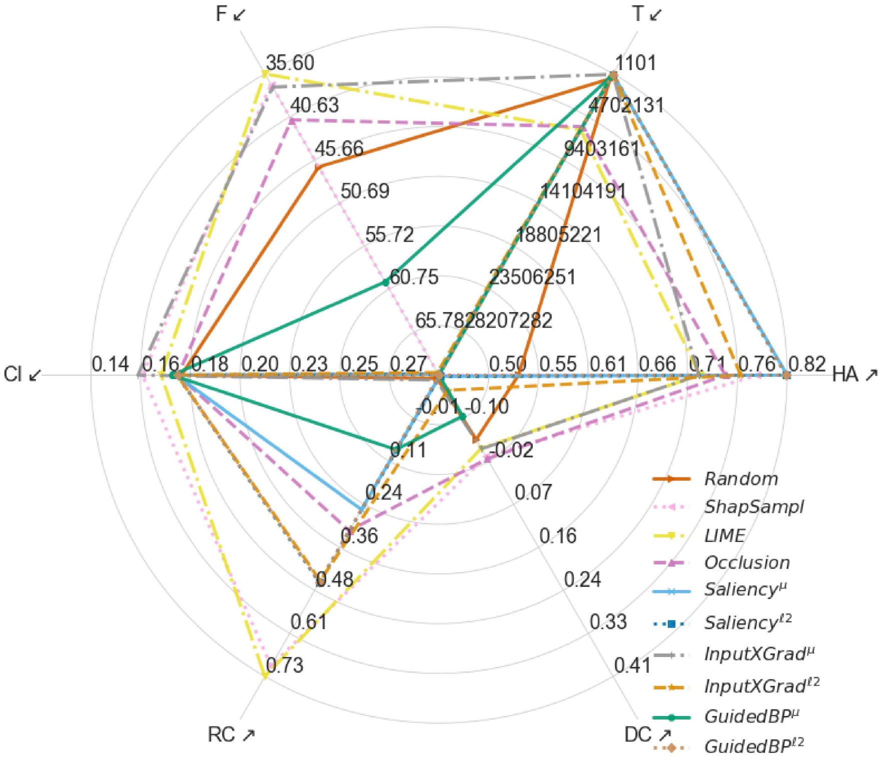


Figure 7.6: Diagnostic property evaluation for all explainability techniques, on the TSE dataset, CNN model. The ↗ and ↙ signs indicate that higher, correspondingly lower, values of the property measure are better.

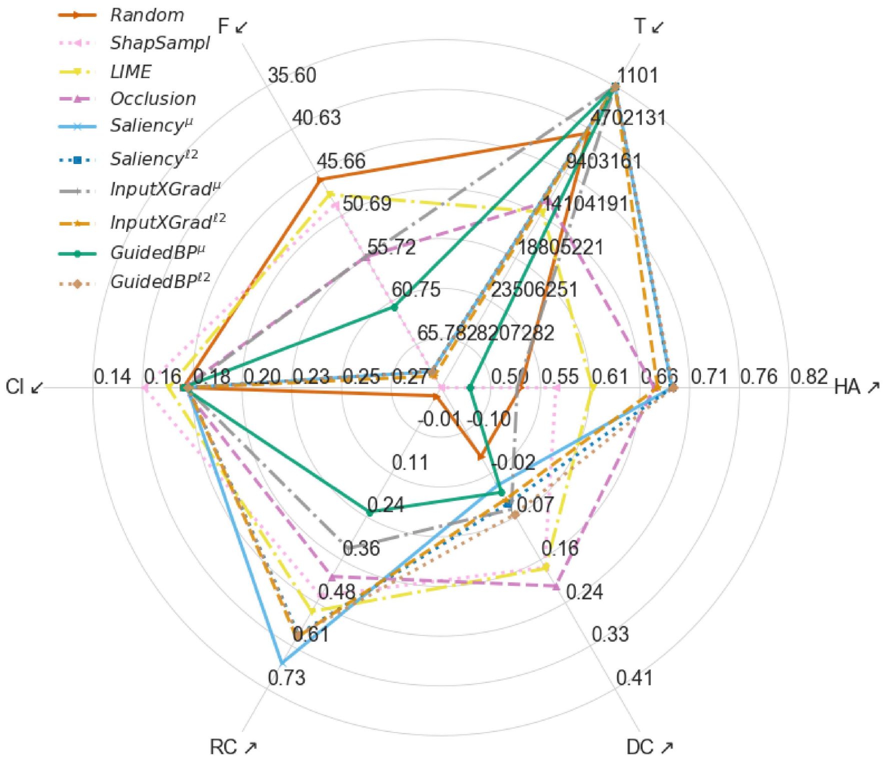


Figure 7.7: Diagnostic property evaluation for all explainability techniques, on the TSE dataset, `LSTM` model. The ↗ and ↙ signs indicate that higher, correspondingly lower, values of the property measure are better.

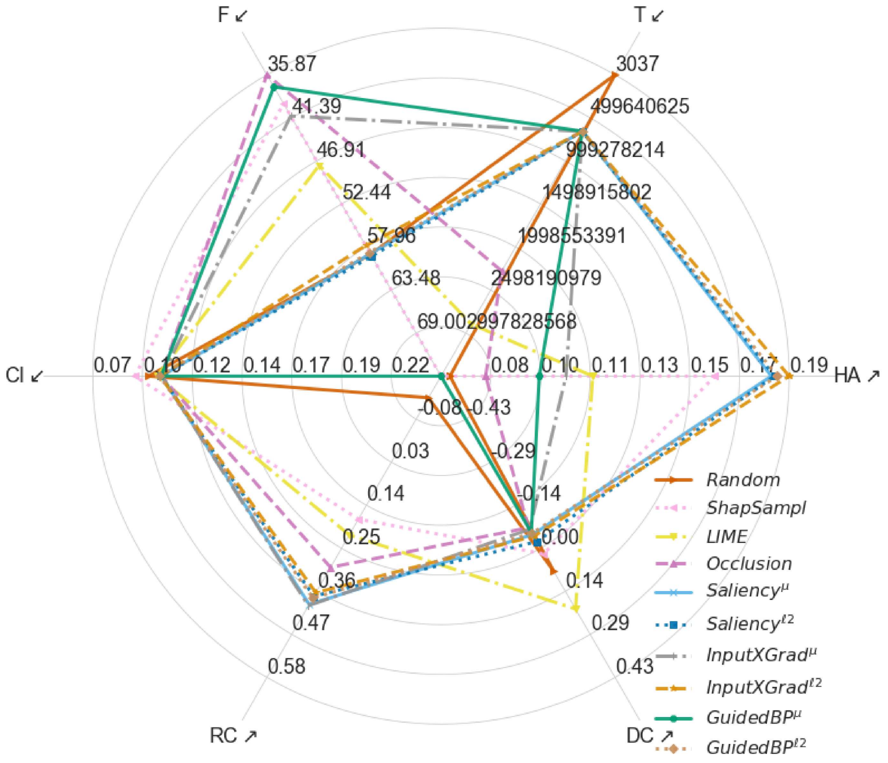


Figure 7.8: Diagnostic property evaluation for all explainability techniques, on the IMDB dataset, `Transformer` model. The ↗ and ↙ signs following the names of each explainability method indicate that higher, correspondingly lower, values of the property measure are better.

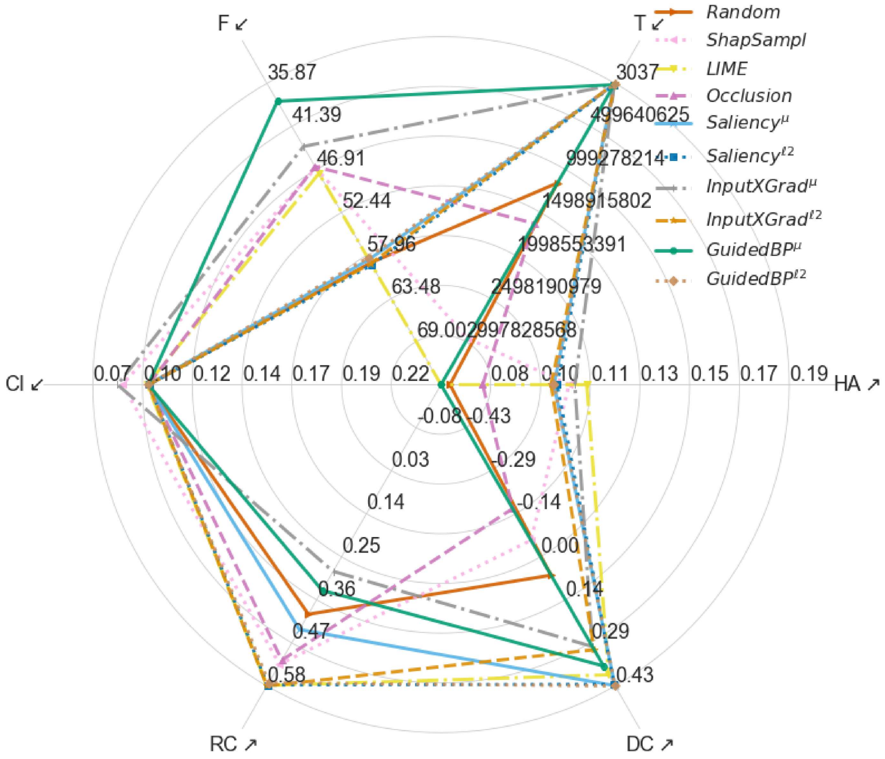


Figure 7.9: Diagnostic property evaluation for all explainability techniques, on the IMDB dataset, CNN model. The ↗ and ↙ signs following the names of each explainability method indicate that higher, correspondingly lower, values of the property measure are better.

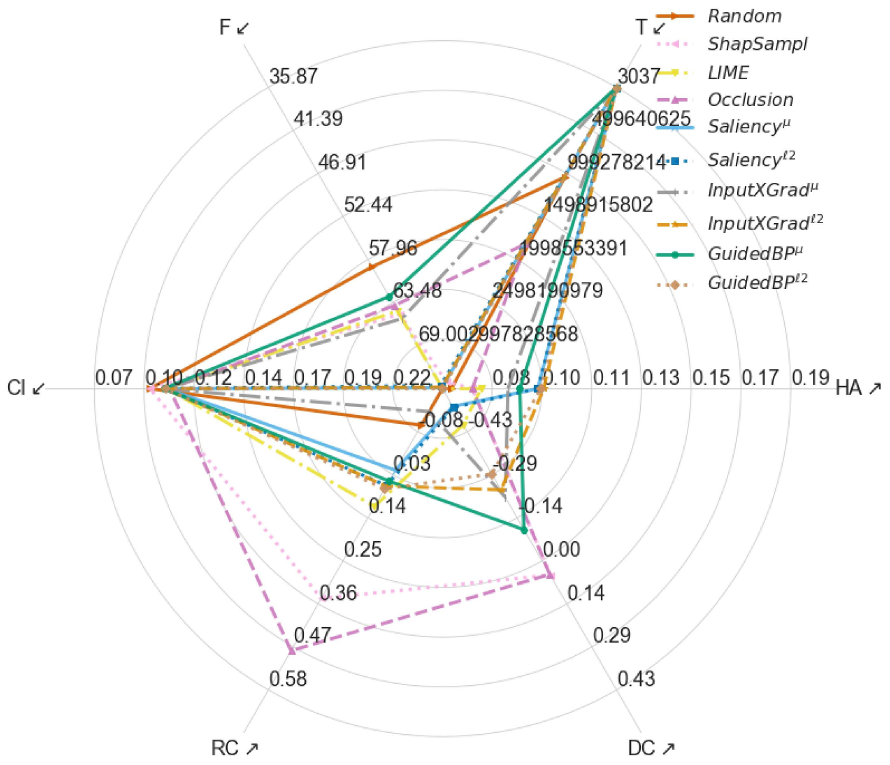


Figure 7.10: Diagnostic property evaluation for all explainability techniques, on the IMDB dataset, `LSTM` model. The ↗ and ↙ signs following the names of each explainability method indicate that higher, correspondingly lower, values of the property measure are better.

Explain.	e-SNLI			IMDB			TSE		
	MAP	MAP RI	FLOPs	MAP	MAP RI	FLOPs	MAP	MAP RI	FLOPs
Random	.297 (±.001)	–	6.12e+3 (±4.6e+1)	.079 (±.001)	–	9.41e+4 (±1.8e+2)	.573 (±.001)	–	4.62e+3 (±2.2e+1)
Transformer									
ShapSampl	.511 (±.004)	.292 (±.011)	1.78e+7 (±5.5e+5)	.168 (±.003)	.084 (±.001)	3.00e+9 (±1.3e+8)	.716 (±.003)	.575 (±.027)	1.29e+7 (±2.0e+6)
LIME	.465 (±.008)	.264 (±.004)	2.39e+5 (±1.5e+4)	.127 (±.004)	.075 (±.004)	4.98e+8 (±1.4e+8)	.745 (±.003)	.570 (±.028)	2.82e+7 (±1.6e+6)
Occlusion	.537 (±.014)	.292 (±.009)	6.33e+5 (±1.0e+3)	.091 (±.001)	.084 (±.001)	8.05e+7 (±4.5e+5)	.710 (±.008)	.577 (±.012)	5.86e+5 (±1.6e+2)
Saliency$^{\mu}$	.614 (±.003)	.255 (±.008)	5.38e+4 (±1.8e+2)	.187 (±.005)	.079 (±.001)	6.59e+5 (±1.8e+3)	.725 (±.011)	.499 (±.002)	4.93e+4 (±2.1e+2)
Saliency$^{\ell 2}$	.615 (±.003)	.255 (±.009)	5.39e+4 (±1.3e+2)	.188 (±.006)	.078 (±.001)	6.62e+5 (±8.4e+2)	.726 (±.014)	.498 (±.001)	4.93e+4 (±1.4e+2)
InputXGrad$^{\mu}$	.356 (±.005)	.280 (±.016)	5.38e+4 (±1.8e+2)	.118 (±.003)	.083 (±.001)	6.60e+5 (±4.5e+3)	.620 (±.008)	.558 (±.011)	4.92e+4 (±1.4e+2)
InputXGrad$^{\ell 2}$	**.624 (±.004)**	.254 (±.013)	5.39e+4 (±1.5e+2)	**.193 (±.005)**	.079 (±.001)	6.62e+5 (±2.1e+3)	**.774 (±.009)**	.499 (±.005)	4.92e+4 (±8.0e+1)
GuidedBP$^{\mu}$	.340 (±.012)	.281 (±.025)	5.39e+4 (±1.8e+2)	.109 (±.003)	.086 (±.005)	6.54e+5 (±7.5e+3)	.589 (±.006)	.567 (±.008)	4.94e+4 (±4.1e+2)
GuidedBP$^{\ell 2}$	.615 (±.003)	.255 (±.009)	5.38e+4 (±1.1e+2)	.189 (±.005)	.079 (±.001)	6.59e+5 (±2.8e+3)	.726 (±.012)	.498 (±.001)	4.97e+4 (±4.2e+2)
CNN									
ShapSampl	.471 (±.003)	.298 (±.008)	3.79e+7 (±3.1e+3)	.119 (±.004)	.084 (±.001)	1.26e+7 (±1.6e+5)	.789 (±.004)	.586 (±.017)	4.53e+6 (±2.1e+4)
LIME	.466 (±.002)	.300 (±.017)	1.81e+4 (±1.2e+3)	**.125 (±.005)**	.079 (±.004)	5.39e+7 (±1.9e+4)	.737 (±.002)	.581 (±.021)	1.52e+4 (±7.1e+1)
Occlusion	.487 (±.003)	.298 (±.006)	6.06e+4 (±2.9e+2)	.090 (±.001)	.084 (±.001)	3.36e+5 (±2.6e+3)	.760 (±.004)	.580 (±.006)	1.40e+4 (±3.6e+1)
Saliency$^{\mu}$	**.600 (±.002)**	.339 (±.007)	1.08e+4 (±5.6e+1)	.114 (±.005)	.091 (±.001)	4.28e+3 (±2.3e+2)	**.816 (±.003)**	.593 (±.008)	4.16e+3 (±1.9e+1)
Saliency$^{\ell 2}$	**.600 (±.002)**	.339 (±.007)	1.06e+4 (±5.6e+1)	.115 (±.005)	.090 (±.001)	4.29e+3 (±9.9e+1)	.815 (±.003)	.596 (±.009)	4.16e+3 (±1.2e+1)
InputXGrad$^{\mu}$	.435 (±.001)	.294 (±.014)	1.07e+4 (±2.3e+1)	.121 (±.003)	.086 (±.002)	4.27e+3 (±1.8e+2)	.736 (±.002)	.572 (±.011)	4.16e+3 (±1.2e+1)
InputXGrad$^{\ell 2}$	.580 (±.001)	.280 (±.003)	1.06e+4 (±6.5e+1)	.113 (±.004)	.093 (±.002)	4.09e+3 (±1.8e+2)	.774 (±.003)	.501 (±.006)	4.12e+3 (±2.7e+1)
GuidedBP$^{\mu}$	.269 (±.001)	.299 (±.017)	1.08e+4 (±1.7e+2)	.076 (±.002)	.086 (±.002)	4.27e+3 (±2.2e+2)	.501 (±.006)	.573 (±.013)	4.32e+3 (±4.0e+2)
GuidedBP$^{\ell 2}$	**.600 (±.002)**	.339 (±.007)	1.07e+4 (±3.4e+1)	.114 (±.005)	.091 (±.002)	4.21e+3 (±2.2e+2)	.815 (±.003)	.594 (±.009)	4.14e+3 (±1.7e+1)
LSTM									
ShapSampl	.396 (±.012)	.291 (±.008)	8.42e+5 (±1.2e+4)	.086 (±.001)	.084 (±.000)	2.30e+8 (±2.5e+5)	.605 (±.034)	.588 (±.020)	1.12e+7 (±2.1e+6)
LIME	.429 (±.012)	.309 (±.018)	1.68e+5 (±2.1e+5)	.089 (±.001)	.081 (±.002)	3.00e+8 (±1.8e+5)	.638 (±.025)	.588 (±.021)	5.20e+4 (±4.1e+3)
Occlusion	.358 (±.003)	.281 (±.007)	2.46e+5 (±5.7e+0)	.086 (±.002)	.083 (±.002)	1.18e+6 (±1.1e+3)	.694 (±.011)	.578 (±.016)	3.71e+4 (±2.7e+0)
Saliency$^{\mu}$	.502 (±.008)	.411 (±.011)	5.11e+3 (±6.8e+0)	.108 (±.001)	.106 (±.000)	3.04e+3 (±7.7e+1)	.710 (±.009)	.546 (±.000)	1.11e+3 (±2.8e+0)
Saliency$^{\ell 2}$	.502 (±.008)	.410 (±.010)	5.12e+3 (±4.6e+0)	.108 (±.002)	.106 (±.002)	3.07e+3 (±3.9e+1)	.710 (±.010)	.546 (±.001)	1.10e+3 (±1.4e+0)
InputXGrad$^{\mu}$	.364 (±.004)	.349 (±.027)	5.12e+3 (±7.2e+0)	.098 (±.002)	.096 (±.002)	3.06e+3 (±7.0e+1)	.570 (±.010)	.601 (±.017)	1.11e+3 (±2.2e+0)
InputXGrad$^{\ell 2}$	**.511 (±.007)**	.389 (±.004)	5.12e+3 (±4.2e+0)	**.110 (±.001)**	.107 (±.000)	3.05e+3 (±9.9e+1)	.697 (±.007)	.544 (±.001)	1.10e+3 (±1.6e+0)
GuidedBP$^{\mu}$	.333 (±.009)	.382 (±.033)	5.11e+3 (±4.4e+0)	.102 (±.005)	.098 (±.003)	3.06e+3 (±1.0e+2)	.527 (±.005)	.570 (±.031)	1.10e+3 (±2.2e+0)
GuidedBP$^{\ell 2}$	.502 (±.009)	.410 (±.009)	5.10e+3 (±2.5e+1)	.109 (±.001)	.107 (±.001)	3.08e+3 (±9.2e+1)	**.711 (±.009)**	.547 (±.001)	1.10e+3 (±2.4e+0)

Table 7.5: Evaluation of the explainability techniques with Human Agreement (HA) and time for computation. HA is measured with Mean Average Precision (MAP) with the gold human annotations, MAP of a Randomly initialized model (MAP RI). The time is computed with FLOPs. The presented numbers are averaged over five different models and the standard deviation of the scores is presented in brackets. Explainability methods with the best MAP for a particular dataset and model are in bold, while the best MAP across all models for a dataset is underlined as well. Methods that have MAP worse than the randomly generated saliency are in red.

Explain.	**e-SNLI**	**IMDB**	**TSE**
Random	56.05 (±0.71)	49.26 (±1.94)	56.45 (±2.37)
		`Transformer`	
ShapSampl	56.05 (±0.71)	65.84 (±11.8)	52.99 (±4.24)
LIME	48.14 (±10.8)	59.04 (±13.7)	42.17 (±7.89)
Occlusion	55.24 (±3.77)	69.00 (±6.22)	52.23 (±4.29)
Saliency$^{\mu}$	37.98 (±2.18)	49.32 (±9.01)	**39.20 (±3.06)**
Saliency$^{\ell 2}$	38.01 (±2.19)	**49.05 (±9.16)**	39.29 (±3.14)
InputXGrad$^{\mu}$	56.98 (±1.89)	64.47 (±8.70)	55.52 (±2.59)
InputXGrad$^{\ell 2}$	**37.05 (±2.29)**	50.22 (±8.85)	37.04 (±2.69)
GuidedBP$^{\mu}$	53.43 (±1.00)	67.68 (±6.94)	57.56 (±2.60)
GuidedBP$^{\ell 2}$	38.01 (±2.19)	49.47 (±8.89)	39.26 (±3.18)
		`CNN`	
ShapSampl	51.78 (±2.24)	59.69 (±8.37)	64.72 (±1.75)
LIME	56.16 (±1.67)	59.09 (±8.48)	65.78 (±1.59)
Occlusion	54.32 (±0.94)	59.86 (±7.78)	61.17 (±1.48)
Saliency$^{\mu}$	34.26 (±1.78)	49.61 (±5.26)	35.70 (±2.94)
Saliency$^{\ell 2}$	34.16 (±1.81)	**49.04 (±5.60)**	35.67 (±2.91)
InputXGrad$^{\mu}$	47.06 (±3.82)	62.05 (±7.54)	64.45 (±2.99)
InputXGrad$^{\ell 2}$	**31.55 (±2.83)**	49.20 (±5.96)	35.86 (±3.22)
GuidedBP$^{\mu}$	47.68 (±2.65)	67.03 (±4.36)	44.93 (±1.57)
GuidedBP$^{\ell 2}$	34.16 (±1.81)	49.80 (±5.99)	**35.60 (±2.91)**
		`LSTM`	
ShapSampl	51.05 (±4.47)	44.05 (±3.06)	53.97 (±6.00)
LIME	51.93 (±7.73)	44.41 (±3.04)	54.95 (±3.19)
Occlusion	54.73 (±3.12)	45.01 (±3.84)	48.68 (±2.28)
Saliency$^{\mu}$	38.29 (±1.77)	35.98 (±2.11)	**37.20 (±3.48)**
Saliency$^{\ell 2}$	38.26 (±1.84)	36.22 (±2.04)	37.23 (±3.50)
InputXGrad$^{\mu}$	49.52 (±1.81)	43.57 (±4.98)	48.71 (±3.23)
InputXGrad$^{\ell 2}$	**37.95 (±2.06)**	36.03 (±1.97)	36.75 (±3.35)
GuidedBP$^{\mu}$	44.48 (±2.12)	46.00 (±3.20)	43.72 (±5.69)
GuidedBP$^{\ell 2}$	38.17 (±1.80)	**35.87 (±1.99)**	37.21 (±3.48)

Table 7.6: Faithfulness-AUC for thresholds $\in$ [0, 10, 20, . . . , 100]. *Lower scores* indicate the ability of the saliency approach to assign higher scores to words more responsible for the final prediction. The scores are mean of different random initializations; the standard deviation is shown in brackets. The smallest AUC for a particular dataset and model are in bold; the smallest AUC across all models for a dataset is underlined. AUC worse than the randomly generated saliency are in red.

Explain.	e-SNLI MAE	 MAX	 MAE-up	 MAX-up	IMDB MAE	 MAX	 MAE-up	 MAX-up	TSE MAE	 MAX	 MAE-up	 MAX-up
Random	.087 (±.004)	.527 (±.007)	.276 (±.005)	.377 (±.002)	.130 (±.007)	.286 (±.014)	.160 (±.003)	.251 (±.008)	.092 (±.009)	.466 (±.021)	.260 (±.017)	.428 (±.064)
						Transformer						
ShapSampl	.071 (±.005)	.456 (±.037)	.158 (±.029)	.437 (±.046)	**.071 (±.008)**	**.238 (±.036)**	**.120 (±.033)**	**.213 (±.035)**	**.073 (±.012)**	**.408 (±.043)**	**.169 (±.052)**	**.415 (±.030)**
LIME	**.068 (±.002)**	**.368 (±.151)**	**.136 (±.028)**	**.395 (±.128)**	.077 (±.008)	.288 (±.024)	.184 (±.018)	.260 (±.021)	.084 (±.009)	.521 (±.072)	.232 (±.013)	.661 (±.225)
Occlusion	.074 (±.004)	.499 (±.020)	.224 (±.006)	.518 (±.048)	.085 (±.011)	.306 (±.015)	.196 (±.015)	.252 (±.011)	.085 (±.011)	.463 (±.035)	.247 (±.015)	.482 (±.091)
Saliency$^{\mu}$	.078 (±.005)	.544 (±.014)	.269 (±.004)	.416 (±.043)	.083 (±.009)	.303 (±.008)	.197 (±.017)	.269 (±.023)	.085 (±.012)	.474 (±.021)	.248 (±.017)	.467 (±.091)
Saliency$^{\ell 2}$	.078 (±.005)	.565 (±.051)	.259 (±.007)	.571 (±.095)	.083 (±.009)	.306 (±.017)	.195 (±.021)	.245(±.004)	.085 (±.012)	.465 (±.021)	.255 (±.012)	.479 (±.074)
InputXGrad$^{\mu}$	.079 (±.005)	.502 (±.015)	.242 (±.006)	.518 (±.031)	.084 (±.011)	.310 (±.011)	.198 (±.013)	.246 (±.008)	.085 (±.011)	.463 (±.015)	.237 (±.010)	.480 (±.071)
InputXGrad$^{\ell 2}$	.078 (±.005)	.568 (±.057)	.258 (±.007)	.581 (±.096)	.083 (±.011)	.301 (±.014)	.193 (±.023)	.249 (±.016)	.086 (±.013)	.469 (±.022)	.252 (±.016)	.480 (±.087)
GuidedBP$^{\mu}$	.080 (±.005)	.505 (±.016)	.242 (±.008)	.519 (±.037)	.084 (±.011)	.308 (±.009)	.196 (±.014)	.245 (±.014)	.085 (±.011)	.456 (±.014)	.237 (±.013)	.494 (±.069)
GuidedBP$^{\ell 2}$	.078 (±.005)	.565 (±.051)	.258 (±.007)	.573 (±.095)	.080 (±.012)	.306 (±.009)	.192 (±.018)	.244 (±.008)	.086 (±.012)	.503 (±.053)	.261 (±.017)	.450 (±.081)
						CNN						
ShapSampl	**.103 (±.001)**	.439 (±.020)	**.133 (±.003)**	.643 (±.032)	.077 (±.018)	.210 (±.041)	.085 (±.023)	.196 (±.026)	.093 (±.002)	**.372 (±.011)**	.148 (±.004)	.479 (±.030)
LIME	.125 (±.003)	.498 (±.018)	.190 (±.006)	.494 (±.028)	.128 (±.006)	.289 (±.019)	.156 (±.003)	.260 (±.011)	.103 (±.001)	.469 (±.027)	.202 (±.014)	.633 (±.090)
Occlusion	.119 (±.004)	.492 (±.018)	.176 (±.007)	.507 (±.037)	.130 (±.007)	.289 (±.018)	.160 (±.006)	.254 (±.005)	.114 (±.002)	.463 (±.018)	.250 (±.007)	.418 (±.035)
Saliency$^{\mu}$	.137 (±.002)	.496 (±.011)	.220 (±.006)	.399 (±.010)	.129 (±.007)	.288 (±.021)	.159 (±.003)	.253 (±.013)	.115 (±.002)	.467 (±.014)	.245 (±.007)	.425 (±.028)
Saliency$^{\ell 2}$	.140 (±.003)	.492 (±.009)	.225 (±.005)	.354 (±.009)	.130 (±.006)	.286 (±.019)	.161 (±.004)	.250 (±.005)	.114 (±.002)	.475 (±.016)	.248 (±.006)	.405 (±.031)
InputXGrad$^{\mu}$	.110 (±.001)	**.436 (±.014)**	.153 (±.007)	.460 (±.009)	**.071 (±.004)**	**.191 (±.010)**	**.071 (±.005)**	**.190 (±.010)**	**.090 (±.002)**	.379 (±.012)	**.135 (±.004)**	.477 (±.025)
InputXGrad$^{\ell 2}$	.140 (±.003)	.492 (±.009)	.225 (±.005)	.355 (±.007)	.130 (±.007)	.285 (±.019)	.160 (±.004)	.251 (±.011)	.114 (±.002)	.475 (±.014)	.248 (±.006)	.416 (±.033)
GuidedBP$^{\mu}$	.140 (±.003)	.485 (±.011)	.225 (±.005)	.367 (±.023)	.129 (±.006)	.286 (±.019)	.159 (±.003)	.253 (±.011)	.114 (±.002)	.462 (±.013)	.234 (±.011)	.441 (±.036)
GuidedBP$^{\ell 2}$	.140 (±.003)	.492 (±.009)	.225 (±.005)	**.353 (±.008)**	.130 (±.007)	.289 (±.018)	.159 (±.004)	.252 (±.011)	.114 (±.002)	.473 (±.015)	.249 (±.006)	**.404 (±.029)**
						LSTM						
ShapSampl	**.118 (±.003)**	.622 (±.035)	**.131 (±.005)**	.648 (±.054)	**.060 (±.018)**	**.279 (±.065)**	**.160 (±.014)**	.277 (±.038)	**.087 (±.007)**	**.433 (±.053)**	**.147 (±.015)**	**.393 (±.029)**
LIME	.127 (±.004)	.512 (±.052)	.145 (±.009)	.490 (±.040)	.069 (±.018)	.300 (±.051)	.209 (±.024)	.267 (±.031)	.090 (±.007)	.667 (±.150)	.218 (±.010)	.864 (±.362)
Occlusion	.147 (±.003)	.579 (±.065)	.172 (±.007)	.593 (±.083)	.069 (±.017)	.304 (±.055)	.216 (±.014)	.324 (±.032)	.099 (±.006)	.509 (±.015)	.259 (±.012)	.723 (±.063)
Saliency$^{\mu}$	.163 (±.002)	.450 (±.008)	.195 (±.008)	.398 (±.031)	.069 (±.018)	.301 (±.051)	.208 (±.026)	**.259 (±.022)**	.101 (±.007)	.518 (±.013)	.271 (±.008)	.469 (±.071)
Saliency$^{\ell 2}$	.163 (±.002)	.448 (±.011)	.195 (±.008)	.399 (±.034)	.070 (±.018)	.299 (±.051)	.206 (±.024)	.263 (±.027)	.101 (±.007)	.523 (±.011)	.273 (±.008)	.441 (±.051)
InputXGrad$^{\mu}$	.161 (±.002)	.454 (±.018)	.193 (±.007)	.502 (±.033)	.066 (±.018)	.295 (±.059)	.201 (±.033)	**.262 (±.014)**	.098 (±.007)	.527 (±.005)	.268 (±.008)	.425 (±.035)
InputXGrad$^{\ell 2}$	.163 (±.002)	**.445 (±.011)**	.195 (±.007)	**.394 (±.029)**	.068 (±.018)	.303 (±.050)	.201 (±.031)	.277 (±.024)	.101 (±.007)	.523 (±.008)	.273 (±.007)	.445 (±.038)
GuidedBP$^{\mu}$	.161 (±.001)	.453 (±.014)	.192 (±.007)	.516 (±.058)	.068 (±.019)	.298 (±.055)	.200 (±.024)	.287 (±.045)	.097 (±.006)	.523 (±.017)	.260 (±.016)	.460 (±.045)
GuidedBP$^{\ell 2}$	.163 (±.002)	.446 (±.010)	.195 (±.007)	.396 (±.042)	.069 (±.017)	.300 (±.050)	.204 (±.024)	.279 (±.025)	.101 (±.007)	.525 (±.010)	.273 (±.007)	.474 (±.051)

Table 7.7: Confidence Indication experiments are measured with the Mean Absolute Error (MAE) of the generated saliency scores when used to predict the confidence of the class predicted by the model and the Maximum Error (MAX). We present the result with and without up-sampling(MAE-up, MAX-up) of the model confidence. The presented measures are an average over the set of models trained from from different random seeds. The standard deviation of the scores is presented in brackets. AVG Conf. is the average confidence of the model for the predicted class. The best results for a particular dataset and model are in bold and the best results across a dataset are also underlined. Lower results are better.

Explain.	**e-SNLI**	**IMDB**	**TSE**
		Transformer	
Random	-0.004 (2.6e-01)	-0.035 (1.4e-01)	0.003 (6.1e-01)
ShapSampl	0.310 (0.0e+00)	0.234 (3.6e-12)	0.259 (0.0e+00)
LIME	**0.519 (0.0e+00)**	0.269 (3.0e-31)	0.110 (2.0e-29)
Occlusion	0.215 (0.0e+00)	0.341 (2.6e-50)	0.255 (0.0e+00)
Saliency$^{\mu}$	0.356 (0.0e+00)	0.423 (3.9e-79)	**0.294 (0.0e+00)**
Saliency$^{\ell 2}$	0.297 (0.0e+00)	0.405 (6.9e-72)	0.289 (0.0e+00)
InputXGrad$^{\mu}$	-0.102 (2.0e-202)	**0.426 (2.5e-80)**	-0.010 (1.3e-01)
InputXGrad$^{\ell 2}$	0.311 (0.0e+00)	0.397 (3.8e-69)	0.292 (0.0e+00)
GuidedBP$^{\mu}$	0.064 (1.0e-79)	-0.083 (4.2e-04)	-0.005 (4.9e-01)
GuidedBP$^{\ell 2}$	0.297 (0.0e+00)	0.409 (1.2e-73)	0.293 (0.0e+00)
		CNN	
Random	-0.003 (4.0e-01)	0.426 (2.6e-106)	-0.002 (7.4e-01)
ShapSampl	0.789 (0.0e+00)	0.537 (1.4e-179)	0.704 (0.0e+00)
LIME	0.790 (0.0e+00)	0.584 (1.9e-219)	**0.730 (0.0e+00)**
Occlusion	0.730 (0.0e+00)	0.528 (2.4e-172)	0.372 (0.0e+00)
Saliency$^{\mu}$	0.701 (0.0e+00)	0.460 (4.5e-126)	0.320 (0.0e+00)
Saliency$^{\ell 2}$	**0.819 (0.0e+00)**	0.583 (4.0e-218)	0.499 (0.0e+00)
InputXGrad$^{\mu}$	0.136 (0.0e+00)	0.331 (1.2e-62)	0.002 (7.5e-01)
InputXGrad$^{\ell 2}$	0.816 (0.0e+00)	**0.585 (8.6e-221)**	0.495 (0.0e+00)
GuidedBP$^{\mu}$	0.160 (0.0e+00)	0.373 (5.5e-80)	0.173 (6.3e-121)
GuidedBP$^{\ell 2}$	**0.819 (0.0e+00)**	0.578 (2.4e-214)	0.498 (0.0e+00)
		LSTM	
Random	0.004 (1.8e-01)	0.002 (9.2e-01)	0.010 (1.8e-01)
ShapSampl	0.657 (0.0e+00)	0.382 (1.7e-63)	0.502 (0.0e-00)
LIME	**0.700 (0.0e+00)**	0.178 (3.3e-14)	0.540 (0.0e-00)
Occlusion	0.697 (0.0e+00)	**0.498 (1.7e-113)**	0.454 (0.0e-00)
Saliency$^{\mu}$	0.645 (0.0e+00)	0.098 (3.1e-05)	**0.667 (0.0e-00)**
Saliency$^{\ell 2}$	0.662 (0.0e+00)	0.132 (1.8e-08)	0.596 (0.0e-00)
InputXGrad$^{\mu}$	0.026 (1.9e-14)	-0.032 (1.7e-01)	0.385 (0.0e-00)
InputXGrad$^{\ell 2}$	0.664 (0.0e+00)	0.133 (1.5e-08)	0.604 (0.0e-00)
GuidedBP$^{\mu}$	0.144 (0.0e+00)	0.122 (2.0e-07)	0.295 (0.0e-00)
GuidedBP$^{\ell 2}$	0.663 (0.0e+00)	0.139 (3.1e-09)	0.598 (0.0e-00)

Table 7.8: Rationale Consistency Spearman's ρ correlation; p-value is in brackets. The best results for a dataset and model are in bold and across a dataset are underlined. Correlation lower that the randomly sampled saliency scores is in red.

Explain.	e-SNLI	IMDB	TSE
		Transformer	
Random	0.047 (2.7e-04)	0.127 (6.6e-07)/	0.121 (2.5e-01)
ShapSampl	0.285 (1.8e-02)	0.078 (5.8e-04)	0.308 (3.4e-36)
LIME	0.372 (3.1e-90)	**0.236 (4.6e-07)**	**0.413 (3.4e-120)**
Occlusion	0.215 (9.6e-02)	0.003 (2.0e-04)	0.235 (7.3e-05)
Saliency$^{\mu}$	0.378 (4.3e-57)	0.023 (4.3e-02)	0.253 (1.4e-20)
Saliency$^{\ell 2}$	0.027 (3.0e-05)	-0.043 (5.6e-02)	0.260 (6.8e-21)
InputXGrad$^{\mu}$	0.319 (3.0e-03)	0.008 (1.2e-01)	0.193 (7.5e-05)
InputXGrad$^{\ell 2}$	0.399 (1.9e-78)	0.028 (2.3e-03)	0.247 (4.9e-17)
GuidedBP$^{\mu}$	0.400 (6.7e-31)	0.017 (1.9e-01)	0.228 (5.2e-09)
GuidedBP$^{\ell 2}$	**0.404 (1.4e-84)**	0.019 (4.3e-04)	0.255 (3.1e-20)
		CNN	
Random	0.018 (2.4e-01)	0.115 (1.8e-04)	0.008 (2.0e-01)
ShapSampl	0.015 (1.8e-01)	-0.428 (5.3e-153)	0.037 (1.4e-01)
LIME	0.000 (4.4e-02)	0.400 (1.4e-126)	0.023 (4.0e-01)
Occlusion	-0.076 (6.5e-02)	-0.357 (1.9e-85)	**0.041 (1.7e-01)**
Saliency$^{\mu}$	0.381 (6.9e-91)	0.431 (1.1e-146)	-0.100 (3.9e-06)
Saliency$^{\ell 2}$	0.391 (1.7e-98)	0.427 (3.5e-135)	-0.100 (3.7e-06)
InputXGrad$^{\mu}$	0.171 (5.1e-04)	0.319 (1.4e-69)	0.024 (3.5e-01)
InputXGrad$^{\ell 2}$	**0.399 (1.0e-93)**	0.428 (1.4e-132)	-0.076 (1.2e-03)
GuidedBP$^{\mu}$	0.091 (7.9e-02)	0.375 (5.7e-109)	-0.032 (1.1e-01)
GuidedBP$^{\ell 2}$	**0.391 (1.7e-98)**	**0.432 (3.5e-140)**	-0.102 (1.7e-06)
		LSTM	
Random	0.018 (3.9e-01)	0.037 (1.8e-01)	0.016 (9.2e-03)
ShapSampl	0.398 (3.5e-81)	0.230 (8.9e-03)	0.205 (2.1e-16)
LIME	**0.415 (1.2e-80)**	0.079 (8.6e-04)	0.207 (4.3e-16)
Occlusion	0.363 (1.1e-37)	**0.429 (7.5e-137)**	**0.237 (2.9e-29)**
Saliency$^{\mu}$	0.158 (1.7e-17)	-0.177 (1.6e-10)	0.065 (5.8e-03)
Saliency$^{\ell 2}$	0.160 (7.5e-19)	-0.168 (2.0e-15)	0.096 (8.2e-03)
InputXGrad$^{\mu}$	0.142 (3.3e-06)	-0.152 (1.2e-14)	0.106 (2.8e-02)
InputXGrad$^{\ell 2}$	0.183 (7.0e-24)	-0.175 (4.7e-17)	0.089 (8.4e-03)
GuidedBP$^{\mu}$	0.163 (1.9e-12)	-0.060 (4.7e-02)	0.077 (1.2e-02)
GuidedBP$^{\ell 2}$	0.169 (1.8e-12)	-0.214 (5.8e-16)	0.115 (4.3e-02)

Table 7.9: Dataset Consistency results with Spearman ρ; p-value is in brackets. The best results for a dataset and model are in bold and across a dataset are underlined. Correlation lower that the randomly sampled saliency scores are in red.

Diagnostics-Guided Explanation Generation

8

8.1 Introduction

Explanations are an important complement to the predictions of a ML model. They unveil the decisions of a model that lead to a particular prediction, which increases user trust in the automated system and can help find its vulnerabilities. Moreover, "The right . . . to obtain an explanation of the decision reached" is enshrined in the European law (Regulation, 2016).

In NLP, research on explanation generation has spurred the release of datasets (Zaidan et al., 2008; Thorne et al., 2018; Khashabi et al., 2018) containing human rationales for the correct predictions of downstream tasks in the form of word- or sentence-level selections of the input text. Such datasets are particularly beneficial for knowledge-intensive tasks (Petroni et al., 2021) with long sentence-level explanations, e.g., question answering and fact-checking, where identifying the required information is an important prerequisite for a correct prediction. They can be used to supervise and evaluate whether a model employs the correct rationales for its predictions (DeYoung et al., 2020a; Thorne et al., 2018; Augenstein, 2021). The goal of this paper is to improve the sentence-level explanations generated for such complex reasoning tasks.

When human explanation annotations are not present, a common approach (Lei et al., 2016; Yu et al., 2019) is to train models that select regions from the input maximising proximity to original task performance which corresponds to the *Faithfulness* property. Atanasova et al. (2020a) propose Faithfulness and other *diagnostic properties* to evaluate different characteristics of explanations. These include *Data Consistency*, which measures the similarity of the explanations between similar instances, and *Confidence Indication*, which evaluates whether the explanation reflects the model's confidence, among others (see Figure 8.1 for an example).

Contributions[1] We present the first method to *learn the aforementioned diagnostic properties in an unsupervised way*, directly optimising for them to improve the quality of generated explanations. We implement a joint task prediction and explanation generation model, which selects rationales at sentence level. Each property can then be included as an additional training objective in the joint model. With experiments

[1] We make an extended version of the manuscript and code available on https://github.com/copenlu/diagnostic-guided-explanations .

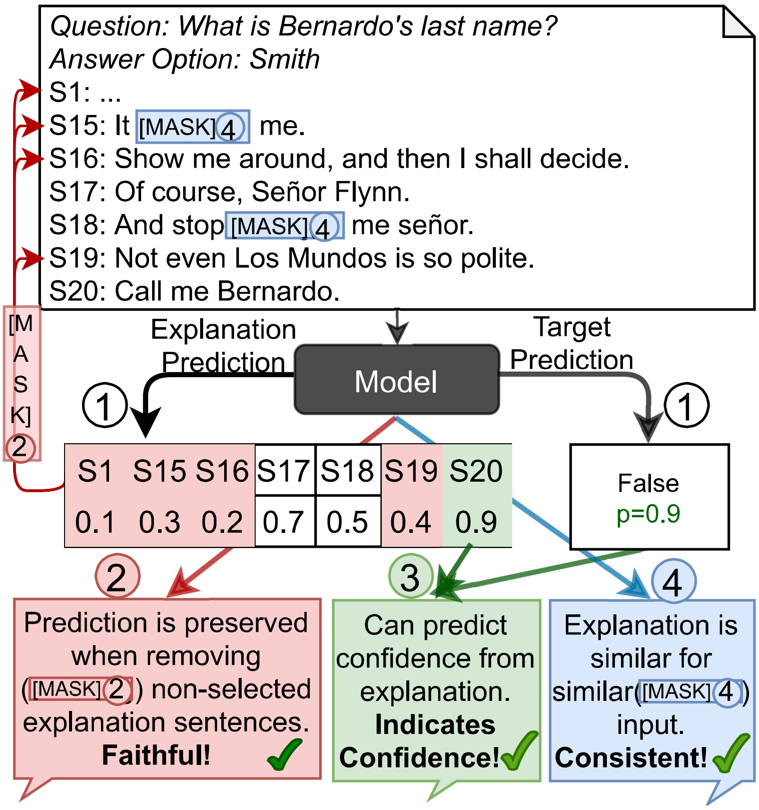

Question: What is Bernardo's last name?
Answer Option: Smith
S1: ...
S15: It [MASK] 4 me.
S16: Show me around, and then I shall decide.
S17: Of course, Señor Flynn.
S18: And stop [MASK] 4 me señor.
S19: Not even Los Mundos is so polite.
S20: Call me Bernardo.
[MASK] 2
Explanation Prediction
Model
Target Prediction
1
S1 S15 S16 S17 S18 S19 S20
0.1 0.3 0.2 0.7 0.5 0.4 0.9
False
p=0.9
2
3
4
Prediction is preserved when removing ([MASK] 2) non-selected explanation sentences. Faithful! ✓
Can predict confidence from explanation. Indicates Confidence! ✓
Explanation is similar for similar([MASK] 4) input. Consistent! ✓

Figure 8.1: Example instance from MultiRC with predicted target and explanation (Step 1), where sentences with confidence ≥ 0.5 are selected as explanations (S17, S18, S20). Steps 2-4 illustrate the use of Faithfulness, Data Consistency, and Confidence Indication diagnostic properties as additional learning signals. '[MASK](2)' is used in Step 2 for sentences (in red) that are not explanations, and '[MASK](4)'–for random words in Step 4.

on three complex reasoning tasks, we find that apart from improving the properties we optimised for, diagnostic-guided training also leads to explanations with higher agreement with human rationales, and improved downstream task performance. Moreover, we find that jointly optimising for diagnostic properties leads to reduced claim/question-only bias (Schuster et al., 2019) for the target prediction, and means that the model relies more extensively on the provided evidence. Importantly, we also find that optimising for diagnostic properties of explanations without supervision for explanation generation does not lead to good human agreement. This indicates the need for human rationales to train models that make the right predictions for the right reasons.

8.2 Related Work

Supervised Explanations. In an effort to guide ML models to perform human-like reasoning and avoid learning spurious patterns (Zhang et al., 2016; Ghaeini et al., 2019), multiple datasets with explanation annotations at the word and sentence level have been proposed (Wiegreffe and Marasovic, 2021). These annotations are also used for supervised explanation generation, e.g., in pipeline models, where the generation task is followed by predicting the target task from the selected rationales only (DeYoung et al., 2020b; Lehman et al., 2019). As Wiegreffe et al. (2021); Kumar and Talukdar (2020); Jacovi and Goldberg (2021) point out, pipeline models produce explanations without task-specific knowledge and without knowing the label to explain. However, for completion, we include the baseline pipeline from ERASER's benchmark DeYoung et al. (2020b) as a reference model for our experiments.

Explanation generation can also be trained jointly with the target task (Atanasova et al., 2020b; Li et al., 2018), which has been shown to improve the performance for both tasks. Furthermore, Wiegreffe et al. (2021) suggest that self-rationalising models, such as multi-task models, provide more label-informed rationales than pipeline models. Such multi-task models can additionally learn a joint probability of the explanation and the target task prediction on the input. This can be decomposed into first extracting evidence, then predicting the class based on it (Zhao et al., 2020; Zhou et al., 2019), or vice versa (Pruthi et al., 2020a). In this work, we also employ joint conditional training. It additionally provides *a good testbed for our experiments with ablations of supervised and diagnostic property objectives, which is not possible with a pipeline approach.*

Most multi-task models encode each sentence separately, then combine their representations, e.g., with Graph Attention Layers (Zhao et al., 2020; Zhou et al., 2019). Glockner et al. (2020) predict the target label from each separate sentence encoding and use the most confident sentence prediction as explanation, which also allows for unsupervised explanation generation. We consider Glockner et al. (2020) as a reference model, as it is the only other work that reports results on generating explanations at the sentence level for three complex reasoning datasets from the ERASER benchmark (DeYoung et al., 2020a). It also outperforms the baseline pipeline model we include from ERASER. *Unlike related multi-task models, we encode the whole input jointly so that the resulting sentence representations are sensitive to the wider document context. The latter proves to be especially beneficial for explanations consisting of multiple sentences. Furthermore, while the model of Glockner et al. (2020) is limited to a fixed and small number (up to two) of sentences per explanation, our model can predict a variable number of sentences depending on the separate instances' rationales.*

Unsupervised Explanations. When human explanation annotations are not provided, model's rationales can be explained with post-hoc methods based on gradients (Sundararajan et al., 2017), simplifications (Ribeiro et al., 2016a), or teacher-student setups (Pruthi et al., 2022). Another approach is to select input tokens that preserve a model's original prediction (Lei et al., 2018; Yu et al., 2019; Bastings et al., 2019; Paranjape et al., 2020), which corresponds to the Faithfulness property of an explanation. However, as such explanations are not supervised by human rationales, they do not have high overlap with human annotations (DeYoung et al., 2020a). Rather, they explain what a model has learned, which does not always correspond to correct rationales and can contain spurious patterns (Wang and Culotta, 2020).

8.3 Method

We propose a novel Transformer (Vaswani et al., 2017) based model to jointly optimise sentence-level explanation generation and downstream task performance. The joint training provides a suitable testbed for our experiments with supervised and diagnostic property objectives for a single model. The joint training optimises two training objectives for the two tasks at the same time. By leveraging information from each task, the model is guided to predict the target task based on correct rationales and to generate explanations based on the model's information needs for target prediction. This provides additional useful information for training each of the tasks. Conducting joint training for these two tasks was shown to improve the performance for each of them (Zhao et al., 2020; Atanasova et al., 2020b).

The **core novelty** is that the model is trained to improve the quality of its explanations by using diagnostic properties of explanations as additional training signals (see Figure 8.1). We select the properties Faithfulness, Data Consistency, and Confidence Indication, as they can be effectively formulated as training objectives. Faithfulness is also employed in explainability benchmarks (DeYoung et al., 2020a) and in related work for unsupervised token-level explanation generation (Lei et al., 2016, 2018), whereas we consider it at sentence level. Further, multiple studies (Yeh et al., 2019; Alvarez-Melis and Jaakkola, 2018) find that explainability techniques are not robust to insignificant and/or adversarial input perturbations, which we address with the Data Consistency property. We do not consider Human Agreement and Rationale Consistency, proposed in Atanasova et al. (2020a). The supervised explanation generation training employs human rationale annotations and thus addresses Human Agreement. Rationale Consistency requires the training of a second model, which is resource-expensive. Another property to investigate in future work is whether a model's prediction can be simulated by another model trained only on the explanations (Hase et al., 2020; Treviso and Martins, 2020; Pruthi et al., 2020b), which also requires training an additional model. We now describe each component in detail.

8.3.1 Joint Modelling

Let $D = \{(x_i, y_i, e_i)|i \in [1, \#(D)]\}$ be a classification dataset. The textual input $x_i = (q_i, a_i^{[opt]}, s_i)$ consists of a question or a claim, an optional answer, and several sentences (usually above 10) $s_i = \{s_{i,j}|j \in [1, \#(s_i)]\}$ used for predicting a classification label $y_i \in [1, N]$. Additionally, D contains human rationale annotations selected from the sentences s_i as a binary vector $e_i = \{e_{i,j} = \{0, 1\}|j \in [1, \#(s_i)]\}$, which defines a binary classification task for explanation extraction.

First, the joint model takes x_i as input and encodes it using a Transformer model, resulting in contextual token representations $h^L = \mathit{encode}(x_i)$ from the final Transformer layer L. From h^L, we select the representations of the CLS token that precedes the question–as it is commonly used for downstream task prediction in Transformer architectures–and the CLS token representations preceding each sentence in s_i, which we use for selecting sentences as an explanation. The selected representations are then transformed with two separate linear layers - h^C for predicting the target, and h^E for generating the explanations, which have the same hidden size as the size of the contextual representations in h^L.

Given representations from h^E, a N-dimensional linear layer predicts the importance $p^E \in \mathbb{R}^{\#(s_i)}$ of the evidence sentences for the prediction of each class. As a final sentence importance score, we only take the score for the predicted class $p^{E[c]}$ and add a sigmoid layer on top for predicting the binary explanation selection task. Given representations from h^C, a N-dimensional linear layer with a soft-max layer on top predicts the target label $p^{C'} \in \mathbb{R}$. The model then predicts the joint conditional likelihood L of the target task and the generated explanation given the input (Eq. 8.1). This is factorised further into first extracting the explanations conditioned on the input and then predicting the target label (Eq. 8.2) based on the extracted explanations (assuming $y_i \perp \mathbf{x}_i \mid \mathbf{e}_i$)).

$$L = \prod_{i=1}^{\#(D)} p\left(y_i, \mathbf{e}_i \mid \mathbf{x}_i\right) \tag{8.1}$$

$$L = \prod_{i=1}^{\#(D)} p\left(\mathbf{e}_i \mid \mathbf{x}_i\right) p\left(y_i \mid \mathbf{e}_i\right) \tag{8.2}$$

We condition the label prediction on the explanation prediction by multiplying $p^{C'}$ and p^E, resulting in the final prediction $p^C \in \mathbb{R}$. The model is trained to optimise jointly the target task cross-entropy loss function ($\mathcal{L}_C$) and the explanation generation cross-entropy loss function ($\mathcal{L}_E$):

$$\mathcal{L} = \mathcal{L}_C(p^C, y) + \mathcal{L}_E(p^{E[c]}, e) \tag{8.3}$$

All loss terms of the diagnostic explainability properties described below are added to $\mathcal{L}$ without additional hyper-parameter weights for the separate loss terms.

8.3.2 Faithfulness (F)

The Faithfulness property guides explanation generation to select sentences preserving the original prediction, (Step 2, Fig. 8.1). In more detail, we take sentence explanation scores $p^{E[c]} \in [0, 1]$ and sample from a Bernoulli distribution the sentences which should be preserved in the input: $c^E \sim \mathit{Bern}(p^{E[c]})$. Further, we make two predictions – one, where only the selected sentences are used as an input for the model, thus producing a new target label prediction l^S, and one where we use only unselected sentences, producing the new target label prediction l^{Co}. The assumption is that a high number $\#(l^{C=S})$ of predictions l^S matching the original l^C indicate the sufficiency (S) of the selected explanation. On the contrary, a low number $\#(l^{C=Co})$ of predictions l^{Co} matching the original l^C indicate the selected explanation is complete (Co) and no sentences indicating the correct label are missed. We then use the REINFORCE (Williams, 1992) algorithm to maximise the reward:

$$\mathcal{R}_F = \#(l^{C=S}) - \#(l^{C=Co}) - |\%(c^E) - \lambda| \tag{8.4}$$

The last term is an additional sparsity penalty for selecting more/less than λ% of the input sentences as an explanation, λ is a hyper-parameter.

8.3.3 Data Consistency (DC)

Data Consistency measures how similar the explanations for similar instances are. Including it as an additional training objective can serve as regularisation for the model to be consistent in the generated explanations. To do so, we mask K random words in the input, where K is a hyper-parameter depending on the dataset. We use the masked text (M) as an input for the joint model, which predicts new sentence scores p^{EM}. We then construct an $\mathcal{L}1$ loss term for the property to minimise for the absolute difference between p^E and p^{EM}:

$$\mathcal{L}_{DC} = |p^E - p^{EM}| \tag{8.5}$$

We use $\mathcal{L}1$ instead of $\mathcal{L}2$ loss as we do not want to penalise for potentially masking important words, which would result in entirely different outlier predictions.

8.3.4 Confidence Indication (CI)

The CI property measures whether generated explanations reflect the confidence of the model's predictions (Step 3, Fig. 8.1). We consider this a useful training objective to re-calibrate and align the prediction confidence values of both tasks. To learn explanations that indicate prediction confidence, we aggregate the sentence importance scores, taking their maximum, minimum, mean, and standard deviation. We transform the four statistics with a linear layer that predicts the confidence $\hat{p}^C$ of the original prediction. We train the model to minimise $\mathcal{L}1$ loss between $\hat{p}^C$ and p^C:

$$\mathcal{L}_{CI} = |p^C - \hat{p}^C| \tag{8.6}$$

We choose $\mathcal{L}1$ as opposed to $\mathcal{L}2$ loss as we do not want to penalise possible outliers due to sentences having high confidence for the opposite class.

8.4 Experiments

8.4.1 Datasets

We perform experiments on three datasets from the ERASER benchmark (DeYoung et al., 2020a) (FEVER, MultiRC, Movies), all of which require complex reasoning and have sentence-level rationales. For FEVER (Thorne et al., 2018), given a claim and an evidence document, a model has to predict the veracity of a claim∈{support, refute}. The evidence for predicting the veracity has to be extracted as explanation. For MultiRC (Khashabi et al., 2018), given a question, an answer option, and a document, a model has to predict if the answer is correct. For Movies (Zaidan et al., 2008), the sentiment∈{positive, negative} of a long movie review has to be predicted. For Movies, as in Glockner et al. (2020), we mark each sentence containing annotated explanation at token level as an explanation. Note that, in knowledge-intensive tasks such as fact checking and question answering also explored here, human rationales point to regions in the text containing the information needed for prediction. Identifying the required information becomes an important preliminary for the correct prediction rather than a plausibility indicator (Jacovi and Goldberg, 2020), and is evaluated as well (e.g., FEVER score, Joint Accuracy).

8.4.2 Metrics

We evaluate the effect of using diagnostic properties as additional training objectives for explanation generation. We first measure their effect on selecting human-like explanations by evaluating precision, recall, and macro F_1 score against human explanation annotations provided in each dataset (§8.5.1). Second, we compute how generating improved explanations affects the target task performance by computing accuracy and macro F_1 score for the target task labels (§8.5.2). Additionally, as identifying the required information in knowledge-intensive datasets, such as FEVER and MultiRC, is an important preliminary for a correct prediction, and following Thorne et al. (2018); Glockner et al. (2020), we evaluate the joint target and explanation performance by considering a prediction as correct only when the whole explanation is retrieved (Acc. Full). In case of multiple possible explanations e_i for one instance (ERASER provides comprehensive explanation annotations for the test sets), selecting one of them counts as a correct prediction. Finally, as diagnostic property training objectives target particular properties, we measure the improvements for each property (§8.5.3).

8.4.3 Experimental Setting

Our core goal is to measure the relative improvement of the explanations generated by the underlying model with (as opposed to without) diagnostic properties. We conduct experiments for the supervised model (Sup.), including separately Faithfulness (F), Data Consistency (DC), and Confidence Indication (CI), as well as all three (All) as additional training signals (§8.3).

Nevertheless, we include results from two other architectures generating sentence-level explanations that serve as a reference for explanation generation performance on the employed datasets. Particularly, we include the best supervised sentence explanation generation results reported in Glockner et al. (2020), and the baseline pipeline model from ERASER, which extracts one sentence as explanation and uses it for target prediction (see §8.2 for a detailed comparison). We also include an additional baseline comparison for the target prediction task. The BERT Blackbox model predicts the target task from the whole document as an input without being supervised by human rationales. The results are as reported by Glockner et al. (2020). In our experiments, we use BERT (Devlin et al., 2019) base-uncased as our base architecture, following Glockner et al. (2020).

8.5 Results

8.5.1 Explanation Generation Results

In Table 8.1, we see that our supervised model performs better than Glockner et al. (2020); DeYoung et al. (2020b). For the MultiRC dataset, where the explanation consists of more than one sentence, our model brings an improvement of more than 30 F_1 points over the reference models, confirming the importance of the contextual information, which performs better than encoding each explanation sentence separately.

When using the diagnostic properties as additional training objectives, we see further improvements in the generated explanations. The most significant improvement is

Method	F_1-C	Acc-C	P-E	R-E	F_1-E	Acc-Joint
FEVER						
Blackbox(Glockner et al., 2020)	90.2 ±0.4	90.2 ±0.4				
Pipeline (DeYoung et al., 2020a)	87.7	87.8	88.3	87.7	88.0	78.1
Supervised (Glockner et al., 2020)	90.7 ±0.7	90.7 ±0.7	92.3 ±0.1	91.6 ±0.1	91.9 ±0.1	83.9 ±0.1
Supervised	89.3 ±0.4	89.4 ±0.3	94.0 ±0.1	93.8 ±0.1	93.9 ±0.1	80.1 ±0.4
Supervised+Data Consistency	**89.7 ±0.5**	**89.7 ±0.5**	**94.4 ±0.0**	**94.2 ±0.0**	**94.4 ±0.0**	80.8 ±0.5
Supervised+Faithfulness	89.5 ±0.4	89.6 ±0.4	92.8 ±0.2	93.7 ±0.2	93.3 ±0.2	75.4 ±0.3
Supervised+Confidence Indication	87.9 ±1.0	87.9 ±1.0	93.9 ±0.1	93.7 ±0.1	93.8 ±0.1	78.5 ±0.9
Supervised+All	89.6 ±0.1	89.6 ±0.1	94.4 ±0.1	94.2 ±0.1	94.3 ±0.1	**80.9 ±0.1**
MultiRC						
Blackbox(Glockner et al., 2020)	67.3 ±1.3	67.7 ±1.6				
Pipeline (DeYoung et al., 2020a)	63.3	65.0	66.7	30.2	41.6	0.0
Supervised (Glockner et al., 2020)	65.5 ±3.6	67.7 ±1.5	65.8 ±0.2	42.3 ±3.9	51.4 ±2.8	7.1 ±2.6
Supervised	71.0 ±0.3	71.4 ±0.3	78.0 ±0.1	78.6 ±0.5	78.3 ±0.1	16.2 ±0.4
Supervised+Data Consistency	**71.7 ±0.6**	**72.2 ±0.7**	**79.9 ±0.4**	79.0 ±0.8	79.4 ±0.5	**19.3 ±0.4**
Supervised+Faithfulness	71.0 ±0.4	71.3 ±0.4	78.2 ±0.1	79.1 ±0.2	78.6 ±0.1	16.1 ±0.5
Supervised+Confidence Indication	70.6 ±0.7	71.1 ±0.6	77.9 ±0.8	78.3 ±0.5	78.1 ±0.5	16.5 ±1.0
Supervised+All	70.5 ±1.6	71.2 ±1.3	79.7 ±1.1	**79.4 ±0.5**	**79.6 ±0.7**	18.8 ±1.6
Movies						
Blackbox(Glockner et al., 2020)	90.1 ±0.3	90.1 ±0.3				
Pipeline (DeYoung et al., 2020a)	86.0	86.0	87.9	60.5	71.7	40.7
Supervised (Glockner et al., 2020)	85.6 ±3.6	85.8 ±3.5	86.9 ±2.5	62.4 ±0.1	72.6 ±0.9	43.9 ±0.6
Supervised	87.4 ±0.4	87.4 ±0.4	79.6 ±0.6	68.9 ±0.5	73.8 ±0.5	59.4 ±0.6
Supervised+Data Consistency	**90.0 ±0.7**	**90.0 ±0.7**	79.5 ±0.1	69.2 ±0.7	74.0 ±0.8	60.8 ±1.7
Supervised+Faithfulness	89.1 ±0.6	89.1 ±0.6	**80.9 ±0.9**	**69.9 ±1.3**	**74.9 ±1.1**	**62.6 ±1.6**
Supervised+Confidence Indication	89.9 ±0.7	89.9 ±0.7	79.7 ±1.4	69.5 ±0.7	74.3 ±1.0	60.1 ±2.6
Supervised+All	89.9 ±0.7	89.9 ±0.7	80.0 ±1.0	69.5 ±1.0	74.4 ±1.0	60.3 ±2.2

Table 8.1: Target task prediction (F_1-C, Accuracy-C) and explanation generation (Precision-E, Recall-E, F_1-E) results (mean and standard deviation over three random seed runs). Last columns measures joint prediction of target accuracy and explanation generation. The property with the best relative improvement over the supervised model is in bold.

achieved with the Data Consistency property for all datasets with up to 2.5 F_1 points over the underlying supervised model. We assume that the Data Consistency objective can be considered as a regularisation for the model's instabilities at the explanation level. The second highest improvement is achieved with the Faithfulness property, increasing F_1 by up to 1 F_1 point for Movies and MultiRC. We assume that the property does not result in improvements for FEVER as it has multiple possible explanation annotations for one instance, which can make the task of selecting one sentence as a complete explanation ambiguous. Confidence Indication results in improvements only on Movies. We conjecture that Confidence Indication is the least related to promoting similarity to human rationales in the generated explanations. Moreover, the re-calibration of the prediction confidence for both tasks possibly leads to fewer prediction changes, explaining the low scores w.r.t. human annotations. We look into how Confidence Indication affects the selected annotations in §8.5.3, and §8.6. Finally, combining all diagnostic property objectives, results in a performance close to the best performing property for each dataset.

8.5.2 Target Prediction Results

In Table 8.1, the Supervised model, without additional property objectives, consistently improves target task performance by up to 4 points in F_1, compared to the two reference models that also generate explanations, except for FEVER, where the models already achieve high results. This can be due to the model encoding all explanation sentences at once, which allows for a more informed prediction of the correct target class. Our model trained jointly with the target task and explanation prediction objective also has similar performance to the BERT Blackbox model and even outperforms it by 4.4 F_1 points for the MultiRC dataset. Apart from achieving high target prediction performance (F_1-C) on the target task, our supervised model also learns which parts of the input are most important for the prediction, which is an important prerequisite for knowledge-intensive tasks.

We see further improvements in downstream task performance when using the diagnostic properties as additional training objectives. Improvements of the generated explanations usually lead to improved target prediction as they are conditioned on the extracted evidence. Here, we again see that Data Consistency steadily improves the target task's performance with up to 2.5 F_1 points. We also see improvements in F_1 with Faithfulness for FEVER and MultiRC. Finally, we find that improvements in Confidence Indication lead to an improvement for target prediction of 2.5 F_1 points for Movies. Combining all objectives, results in performance close the performance of the other properties.

We also show joint prediction results for target task and evidence. For MultiRC and Movies, the improvements of our supervised model over Glockner et al. (2020) are very considerable with up to 9 accuracy points; using diagnostic properties increases results further up to 4 points in accuracy. Apart from improving the properties of the generated explanations, this could be due to the architecture conditioning the prediction on the explanation. The only dataset we do not see improvements for is FEVER, where again the performance is already high, and the target prediction of our model performs worse than Glockner et al. (2020).

8.5.3 Explanations Property Results

So far, we concentrate on the relative performance improvements compared to human annotations. However, the diagnostic properties' additional training objectives are directed at generating explanations that exhibit these properties to a larger degree. Here, we demonstrate the improvements over the explanation properties themselves for unseen instances in the test splits. Note that this is a control experiment as we expect the properties we optimise for to be improved.

Dataset	Method	Suff. ↑	Compl. ↓
FEVER	Supervised	85.1	85.1
	Supervised+F	97.4	83.6
MultiRC	Supervised	81.7	69.2
	Supervised+F	82.3	67.0
Movies	Supervised	94.8	92.2
	Supervised+F	96.6	91.3

Table 8.2: Sufficiency and Completeness as proportions of the instances that preserve their prediction when evaluated on only the selected (Suff.) or the unselected (Compl.) explanation sentences, accordingly, for training with and without the Faithfulness objective.

Dataset	Method	Pred.	Expl.
FEVER	Sup.	0.03 (9.9e-8)	3.68 (1.80)
	Sup.+DC	0.02 (9.1e-8)	2.56 (0.97)
MultiRC	Sup.	0.09 (5.6e-8)	7.83(2.87)
	Sup.+DC	0.05 (4.9e-8)	3.01(0.89)
Movies	Sup.	0.04 (7.1e-8)	2.34 (1.38)
	Sup.+DC	0.01 (6.2e-8)	1.72 (0.90)

Table 8.3: Mean and standard deviation (in brackets) of the difference between target (Pred.) and explanation (Expl.) prediction confidence for similar (masked) instances.

Faithfulness. In Table 8.2 we see that supervision from the Faithfulness property leads to generating explanations that preserve the original label of the instance for all datasets. For FEVER, the label is even preserved in 12% of the instances more than with the supervised objective only. The least faithful explanations are those generated for MultiRC, which can be explained by the low joint performance of both tasks. We also see that even when removing the selected explanations, it is still possible to predict the same label based on the remaining evidence. Such cases are decreased when including the Faithfulness property. The latter phenomenon can be explained by the fact that FEVER and Movies' instances contain several possible explanations. We conjecture that this might also be due to the model learning spurious correlations. We further study this in Sec. 8.6.1.

Data Consistency. Using Data Consistency as an additional training objective aims to regularise the model to select similar explanations for similar instances. In Table 8.3, we find the variance of downstream task prediction confidence decreases for all datasets with up to 0.04 points. Furthermore, the variance of generated explanation probabilities for similar instances is decreased as well. The largest improvements

Method	FEVER	MultiRC	Movies
Sup.	0.10 (0.17)	0.05 (0.10)	0.12 (0.09)
Sup.+CI	0.05 (0.09)	0.04 (0.09)	0.05 (0.10)

Table 8.4: Mean and standard deviation (in brackets) difference between the model's confidence and the confidence of the generated explanations.

are for MultiRC and Movies, where the property brings the highest performance improvement w.r.t. human annotations as well. We also find that the Movies dataset, which has the longest inputs, has the smallest variance in explanation predictions. This suggests that the variance in explanation prediction is more pronounced for shorter inputs as in FEVER and MultiRC, where the property brings more improvement w.r.t. human annotations. The variance could also depend on the dataset's nature.

Confidence Indication. Table 8.4 shows the difference between the confidence of the predicted target label and the confidence of the explanation sentence with the highest importance. Including Confidence Indication as a training objective indeed decreases the distance between the confidence of the two tasks, making it easier to judge the confidence of the model only based on the generated explanation's confidence. The confidence is most prominently improved for the Movies dataset, where it is also the dataset with the largest improvements for supervised explanation generation with Confidence Indication objective.

8.5.4 Unsupervised Rationale Generation

We explore how well explanations can be generated without supervision from human explanation annotations. Table 8.5 shows that the performance of the unsupervised rationales is limited with an up to 47 F_1 point decrease for FEVER compared to the supervised model. We assume that as our model encodes the whole input together, this leads to a uniform importance of all sentences as they share information through their context. While joint encoding improves the target prediction for complex reasoning datasets especially with more than one explanation sentence, this also limits the unsupervised learning potential of our architecture. As the model is not supervised to select explanations close to human ones, improving the diagnostic properties has a limited effect in improving the results w.r.t. human annotations.

8.6 Discussion

Method	FEVER	MultiRC	Movies
Sup.	93.9 ±0.1	78.3 ±0.1	73.8 ±0.5
UnS.	56.1 ±0.4	34.8 ±7.6	50.0 ±1.8
UnS.+DC	46.9 ±0.4	38.1 ±3.2	63.8 ±1.2
UnS.+F	51.6 ±0.3	24.4 ±5.2	64.6 ±0.4
UnS.+CI	57.5 ±0.4	25.4 ±3.4	60.0 ±1.6
UnS.+All	57.3 ±0.2	37.4 ±6.4	63.6 ±0.3

Table 8.5: Performance on the explanation generation task without human annotation supervision (UnS.).

Dataset	Method	F_1-C	Acc-C
FEVER	Random	26.1 ±4.3	37.1 ±5.6
	Sup.	75.6 ±0.3	75.7 ±0.3
	Sup.+DC	68.2 ±0.2	75.6 ±0.3
	Sup.+F	73.4 ±0.4	73.9 ±0.3
	Sup.+CI	73.2 ±0.4	73.7 ±0.4
	Sup.+All	73.5 ±0.2	73.8 ±0.4
	Sup. on whole input	89.3 ±0.4	89.4 ±0.3
MultiRC	Random	26.1 ±5.5	31.6 ±5.9
	Sup.	59.4 ±0.8	63.5 ±0.9
	Sup.+DC	54.5 ±0.9	61.3 ±1.2
	Sup.+F	57.8 ±0.8	61.4 ±0.6
	Sup.+CI	49.7 ±0.8	60.1 ±0.2
	Sup.+All	59.0 ±0.3	61.0 ±0.2
	Sup. on whole input	71.0 ±0.3	71.4 ±0.3

Table 8.6: Performance of the models for the downstream task when provided with the query-answer part only.

8.6.1 Question/Claim Only Bias

Prior work has found that models can learn spurious correlations between the target task and portions of the input text, e.g., predicting solely based on the claim to be fact checked (Schuster et al., 2019), regardless of the provided evidence. In our experiments, the input for FEVER and MultiRC also contains two parts - a claim or a question-answer pair and evidence text, where the correct prediction of the target always depends on the evidence. Suppose the models do not consider the second part of the input when predicting the target task. In that case, efforts to improve the generated explanations will not affect the target task prediction as it does not rely on that part of the input.

Table 8.6 shows target task performance of models trained on the whole input, but using only the first part of the input at test time. We find that, given the limited input,

Question: What colors are definitely used in the picture Lucy drew?; **Answer:** Yellow and purple; **Label:** True **Predicted:** **Sup** True, p=.98; **Sup+DC** True, p=.99 **E-Sup:** She draws a picture of her family. She makes sure to draw her mom named Martha wearing a purple dress, because that is her favorite. She draws many yellow feathers for her pet bird named Andy. **E-Sup+S:** She makes sure to draw her mom named Martha wearing a purple dress, because that is her favorite. She draws many yellow feathers for her pet bird named Andy.
Claim: Zoey Deutch did not portray Rosemarie Hathaway in Vampire Academy.; **Label:** REFUTE **Predicted:** **Sup** refute, p=.99; **Sup+F** refute, p=.99 **E-Sup:** Zoey Francis Thompson Deutch (born November 10, 1994) is an American actress. **E-Sup+F:** She is known for portraying Rosemarie “Rose” Hathaway in Vampire Academy(2014), Beverly in the Richard Link later film Everybody Wants Some!!
E-Sup/E-Sup+CI: For me, they calibrated my creativity as a child; they are masterful, original works of art that mix moving stories with what were astonishing special effects at the time (and they still hold up pretty well).; **Label:** Positive **Predicted:** **Sup** negative, p=.99 **Sup+CI** positive, p=.99

Table 8.7: Example explanation predictions changed by including the diagnostic properties as training objectives.

the performance is still considerable compared to a random prediction. For FEVER, the performance drops only with 14 F_1 score points to 75.6 F_1 score. This could explain the small relative improvements for FEVER when including diagnostic properties as training objectives, where the prediction does not rely on the explanation to a large extent.

Another interesting finding is that including diagnostic properties as training objectives decreases models’ performance when a supporting document is not provided. We assume this indicates the properties guide the model to rely more on information in the document than to learn spurious correlations between the question/claim and the target only. The Data Consistency and Confidence Indication property lead to the largest decrease in model’s performance on the limited input. This points to two potent objectives for reducing spurious correlations.

8.6.2 Explanation Examples

Table 8.7 illustrates common effects of the diagnostic properties. We find Data Consistency commonly improves explanations by removing sentences unrelated to the target prediction, as in the first example from MultiRC. This is particularly useful for MultiRC, which has multiple gold explanation sentences. For FEVER and Movies, where one sentence is needed, the property brings smaller improvements w.r.t. human explanation annotations.

The second example from FEVER illustrates the effect of including Faithfulness as an objective. Naturally, for instances classified correctly by the supervised model, their generated explanation is improved to reflect the rationale used to predict the target. However, when the prediction is incorrect, the effect of the Faithfulness property is limited.

Finally, we find Confidence Indication often re-calibrates the prediction probabilities of generated explanations and predicted target tasks, which does not change many target predictions. This explains its limited effect as an additional training objective. The re-calibration also influences downstream task prediction confidence, as in the last example from the Movies dataset. This is a side effect of optimising the property while training the target task, where both explanation and target prediction confidence can be changed to achieve better alignment.

8.7 Conclusion

In this paper, we study the use of diagnostic properties for improving the quality of generated explanations. We find that including them as additional training objectives improves downstream task performance and generated explanations w.r.t. human rationale annotations. Moreover, using only the diagnostic properties as training objectives does not lead to a good performance compared to only using human rationale annotations. The latter indicates the need for human rationale annotations for supervising a model to base its predictions on the correct rationales. In future, we plan to experiment with application tasks with longer inputs, where current architectures have to be adjusted to make it computationally possible to encode longer inputs.

Acknowledgments

The research documented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199. Isabelle Augenstein's research is further partially funded by a DFF Sapere Aude research leader grant.

References

Abubakar Abid, Mert Yuksekgonul, and James Zou. 2022. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 66–88. PMLR.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9525–9536, USA. Curran Associates Inc.

Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. 2022. Post hoc Explanations may be Ineffective for Detecting Unknown Spurious Correlation. In *International Conference on Learning Representations*.

Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. 2019. Protecting World Leaders Against Deep Fakes. In *CVPR Workshops*, pages 38–45.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium. Association for Computational Linguistics.

Liesbeth Allein, Isabelle Augenstein, and Marie-Francine Moens. 2021. Time-aware evidence ranking for fact-checking. *Journal of Web Semantics*, 71:100663.

David Alvarez-Melis and Tommi S. Jaakkola. 2018. On the Robustness of Interpretability Methods. *CoRR*, abs/1806.08049.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. 2022. Finding and removing Clever Hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2022. Machine Bias. *Ethics of Data and Analytics: Concepts and Cases*, page 254.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy. Association for Computational Linguistics.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82 – 115.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. In *Proceedings of ICLR*.

Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness. *Central Europe Workshop (CEUR)*.

Pepa Atanasova, Georgi Karadzhov, Yasen Kiprov, Preslav Nakov, and Fabrizio Sebastiani. 2019a. Evaluating Variable-Length Multiple-Option Lists in Chatbots and Mobile Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 997–1000, New York, NY, USA. Association for Computing Machinery.

Pepa Atanasova, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. 2019b. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness. *CLEF (Working Notes)*, 2380.

Pepa Atanasova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Georgi Karadzhov, Tsvetomila Mihaylova, Mitra Mohtarami, and James Glass. 2019c. Automatic Fact-Checking Using Context and Discourse Information. *J. Data and Information Quality*, 11(3).

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020a. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020b. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2021. Diagnostics-Guided Explanation Generation. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'21. AAAI Press.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2022. Fact Checking with Insufficient Evidence. *Transactions of the Association for Computational Linguistics*, 10:746–763.

Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020c. Generating label cohesive and well-formed adversarial claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3168–3177, Online. Association for Computational Linguistics.

Isabelle Augenstein. 2021. *Towards Explainable Fact Checking*. Dr. Scient. thesis, University of Copenhagen, Faculty of Science.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 21–27, New Orleans, Louisiana. Association for Computational Linguistics.

Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the GRU: Multi-Task Learning for Deep Text Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 107–114, New York, NY, USA. Association for Computing Machinery.

Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *California law review*, pages 671–732.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Nathaniel Berger, Stefan Riezler, Sebastian Ebert, and Artem Sokolov. 2021. Don't search for a search method — simple heuristics suffice for adversarial text attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8216–8224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ann Bies, Justin Mott, and Colin Warner. 2015. English News Text Treebank: Penn Treebank Revised. *Web Download*.

Kersti Börjars and Kate Burridge. 2019. *Introducing English Grammar*. Routledge.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman and George Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855, Online. Association for Computational Linguistics.

Keith Brown, Jim Miller, and James Edward Miller. 1991. *Syntax: A Linguistic Introduction to Sentence Structure*. Psychology Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Noel Burton-Roberts. 2016. *Analysing sentences: An introduction to English syntax*. Routledge.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA. Association for Computing Machinery.

Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial Calculation of the Shapley Value Based on Sampling. *Comput. Oper. Res.*, 36(5):1726–1730.

Tuhin Chakrabarty, Tariq Alhindi, and Smaranda Muresan. 2018. Robust document retrieval and individual evidence modeling for fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 127–131, Brussels, Belgium. Association for Computational Linguistics.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45.

Aaron Chan, Shaoliang Nie, Liang Tan, Xiaochang Peng, Hamed Firooz, Maziar Sanjabi, and Xiang Ren. 2022. FRAME: Evaluating Simulatability Metrics for Free-Text Rationales.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4026–4032, Minneapolis, Minnesota. Association for Computational Linguistics.

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4511–4526, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Sagnik Ray Choudhury, Nikita Bhutani, and Isabelle Augenstein. 2021. Can Edge Probing Tasks Reveal Linguistic Knowledge in QA Models?

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Luna De Bruyne, Pepa Atanasova, and Isabelle Augenstein. 2022. Joint emotion label space modeling for affect lexica. *Computer Speech & Language*, 71:101257.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020a. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C. Wallace. 2020b. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 123–132, Online. Association for Computational Linguistics.

Ousmane Amadou Dia, Elnaz Barshan, and Reza Babanezhad. 2019. Semantics Preserving Adversarial Learning. *arXiv preprint arXiv:1903.03905*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. 2020. e-SNLI-VE: Corrected Visual-Textual Entailment with Natural Language Explanations. In *2020 CVPR workshop on Fair, Data Efficient and Trusted Computer Vision*. arXiv.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv preprint arXiv:1704.05179*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Jessica Zosa Forde, Charles Lovering, George Konidaris, Ellie Pavlick, and Michael L Littman. 2022. Where, When & Which Concepts Does AlphaZero Learn? Lessons from the Game of Hex. In *AAAI Workshop on Reinforcement Learning in Games*, volume 2.

Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Hang Gao and Tim Oates. 2019. Universal Adversarial Perturbation for Text Classification. *arXiv preprint arXiv:1910.04618*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay

Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Paul L Garvin. 1958. Syntactic units and operations. In *Proc. VIIIth. Intern. Congress of linguists at Oslo*, pages 58–59. De Gruyter Mouton.

Wendong Ge, Jin-Won Huh, Yu Rang Park, Jae-Ho Lee, Young-Hak Kim, and Alexander Turchin. 2018. An Interpretable ICU Mortality Prediction Model Based on Logistic Regression and Recurrent Neural Networks with LSTM units. In *AMIA Annual Symposium Proceedings*, volume 2018, page 460. American Medical Informatics Association.

Reza Ghaeini, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Saliency Learning: Teaching the Model Where to Pay Attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4016–4025, Minneapolis, Minnesota. Association for Computational Linguistics.

Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind Picard. 2022. DISSECT: Disentangled simultaneous explanations via concept traversals. In *International Conference on Learning Representations*.

Max Glockner, Ivan Habernal, and Iryna Gurevych. 2020. Why do you think that? exploring faithful sentence-level rationales without supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1080–1095, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples.

Bryce Goodman and Seth Flaxman. 2017. European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv preprint arXiv:1706.02677*.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a Deep and Unified Understanding of Deep Neural Models in NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2454–2463, Long Beach, California, USA. PMLR.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895, Melbourne, Australia. Association for Computational Linguistics.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

Casper Hansen, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural Check-Worthiness Ranking With Weak Supervision: Finding Sentences for Fact-Checking. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 994–1000.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. A survey on stance detection for mis-and disinformation identification. *arXiv preprint arXiv:2103.00242*.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

Trevor J Hastie. 2017. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the Call for a Standard Reliability Measure for Coding Data. *Communication methods and measures*, 1(1):77–89.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606, Online. Association for Computational Linguistics.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Rodney Huddleston and Geoffrey Pullum. 2005. The Cambridge grammar of the English language. *Zeitschrift für Anglistik und Amerikanistik*, 53(2):193–194.

Filip Ilievski, Pedro Szekely, and Bin Zhang. 2021. CSKG: The CommonSense Knowledge Graph. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings*, page 680–696, Berlin, Heidelberg. Springer-Verlag.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2021. Aligning Faithful Interpretations with their Social Attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. TextFool: Fool your Model with Natural Adversarial Text.

Ulf Johansson, Rikard König, and Lars Niklasson. 2004. The Truth is in There Rule Extraction from Opaque Models Using Genetic Programming. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*. AAAI Press.

Shailza Jolly, Pepa Atanasova, and Isabelle Augenstein. 2021. Generating Fluent Fact Checking Explanations with Unsupervised Post-Editing. *CoRR*, abs/2112.06924.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. 2018. Multi-source multi-class fake news detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1546–1557, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning The Difference That Makes A Difference With Counterfactually-Augmented Data. In *International Conference on Learning Representations*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Ji-Seong Kim and Key-Sun Choi. 2021. Fact Checking in Knowledge Graphs by Logical Consistency. Preprint on webpage at `http://www.semantic-web-journal.net/system/files/swj2721.pdf`.

Jiho Kim, Kijong Han, and Key-Sun Choi. 2018. KBCNN: A Knowledge Base Completion Model Based On Convolutional Neural Networks. In *Annual Conference on Human and Language Technology*, pages 465–469. Human and Language Technology.

Youngwoo Kim and James Allan. 2019. FEVER breaker's run of team NbAuzDrLqg. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 99–104, Hong Kong, China. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *ArXiv*, abs/1611.07270.

Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, New Orleans, Louisiana. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Neema Kotonya and Francesca Toni. 2020a. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020b. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Sarah Kreps, R Miles McCain, and Miles Brundage. 2022. All the news that's fit to fabricate: AI-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, 9(1):104–117.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2021. ProoFVer: Natural Logic Theorem Proving for Fact Verification. *CoRR*, abs/2108.11357.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Anja Lambrecht and Catherine Tucker. 2019. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. *Management science*, 65(7):2966–2981.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

LanguageTool. 2022. LanguageTool. `https://languagetool.org/`.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, pages 1078–1088. PMLR.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.

Kai Lei, Daoyuan Chen, Yaliang Li, Nan Du, Min Yang, Wei Fan, and Ying Shen. 2018. Cooperative denoising for distantly supervised relation extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 426–436, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Markus Leippold and Thomas Diggelmann. 2020. Climate-FEVER: A Dataset for Verification of Real-World Climate Claims. In *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning*.

Piyawat Lertvittayakumjorn and Francesca Toni. 2019. Human-grounded evaluations of explanation methods for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5195–5205, Hong Kong, China. Association for Computational Linguistics.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. 2020. Unsupervised Text Generation by Learning from Search. In *Advances in Neural Information Processing Systems*, volume 33, pages 10820–10831, Online. Curran Associates, Inc.

Sizhen Li, Shuai Zhao, Bo Cheng, and Hao Yang. 2018. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 138–144, Brussels, Belgium. Association for Computational Linguistics.

Jianhua Lin. 1991. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information theory, 37(1):145–151*, pages 145–151.

Peter Lipton. 1990. Contrastive Explanation. *Royal Institute of Philosophy Supplement*, 27:247–266.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020a. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020b. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020c. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake News Detection Through Multi-Perspective Speaker Profiles. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 252–256.

Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *arXiv preprint arXiv:1711.05101*.

Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect Rumor and Stance Jointly by Neural Multi-task Learning. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 585–593, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Zhuang Ma and Michael Collins. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3698–3707, Brussels, Belgium. Association for Computational Linguistics.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate Your Counterfactuals: Towards Controlled Counterfactual Generation for Text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13516–13524.

Christopher Malon. 2018. Team papelo: Transformer networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113, Brussels, Belgium. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

David Martens, Johan Huysmans, Rudy Setiono, Jan Vanthienen, and Bart Baesens. 2008. *Rule Extraction from Support Vector Machines: An Overview of Issues and Application in Credit Scoring*, pages 33–63. Springer Berlin Heidelberg, Berlin, Heidelberg.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. 2021. Is sparse attention more interpretable? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 122–129, Online. Association for Computational Linguistics.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

Rahul Mishra, Dhruv Gupta, and Markus Leippold. 2020. Generating fact checking summaries for web claims. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 81–90, Online. Association for Computational Linguistics.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 767–776, New Orleans, Louisiana. Association for Computational Linguistics.

Christoph Molnar. 2022. *Interpretable Machine Learning*, 2 edition. Independently published (February 28, 2022).

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Maximilian Mozes, Max Bartolo, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Contrasting human- and machine-generated word-level adversarial examples for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8258–8270, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3075–3081. AAAI Press.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. WT5?! Training Text-to-Text Models to Explain their Predictions. *CoRR*, abs/2004.14546.

Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682*.

Michael Neely, Stefan F. Schouten, Maurits J. R. Bleeker, and Ana Lucic. 2021. Order in the Court: Explainable AI Methods Prone to Disagreement. *CoRR*, abs/2105.03287.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Piotr Niewinski, Maria Pszona, and Maria Janicka. 2019. GEM: Generative enhanced model for adversarial attacks. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 20–26, Hong Kong, China. Association for Computational Linguistics.

Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. 2020. Bias in data-driven artificial intelligence systems—An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In *arXiv:2202.06671*.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. Multi-Hop Fact Checking of Political Claims. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3892–3898. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Frank Pasquale. 2015. *The black box society: The secret algorithms that control money and information*. Harvard University Press.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of machine Learning research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Melbourne, Australia. Association for Computational Linguistics.

Richard D Powers, William A Sumner, and Bryant E Kearl. 1958. A recalculation of four adult readability formulas. *Journal of Educational Psychology*, 49(2):99.

Nicoletta Prentzas, Andrew Nicolaides, Efthyvoulos Kyriacou, Antonis Kakas, and Constantinos Pattichis. 2019. Integrating machine learning with symbolic reasoning to build an explainable AI model for stroke prediction. In *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 817–821. IEEE.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2022. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Danish Pruthi, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020a. Weakly- and semi-supervised evidence extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3965–3970, Online. Association for Computational Linguistics.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020b. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Weizhu Chen, and Jiawei Han. 2021. CoDA: Contrast-enhanced and Diversity-promoting Data Augmentation for Natural Language Understanding. In *International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes.

2020. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 33–44.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. 2022. Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–397.

General Data Protection Regulation. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Nils Rethmeier and Isabelle Augenstein. 2021. A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned and Perspectives. *arXiv preprint arXiv:2102.12982*.

Nils Rethmeier and Isabelle Augenstein. 2022. Long-Tail Zero and Few-Shot Learning via Contrastive Pretraining on and for Small Data. In *Proceedings of AAAI 2022 Workshop on Artificial Intelligence with Biased or Scarce Data (AIBSD 2022)*.

Marco Tulio Ribeiro, UW EDU, Sameer Singh, and Carlos Guestrin. 2016a. Model-Agnostic Interpretability of Machine Learning. In *ICML Workshop on Human Interpretability in Machine Learning*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Michael L Rich. 2016. Machine learning, automated suspicion algorithms, and the fourth amendment. *University of Pennsylvania Law Review*, pages 871–929.

Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In *Human and machine learning*, pages 159–175. Springer.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2021. SOLID: A large-scale semi-supervised dataset for offensive language identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 915–928, Online. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent Multi-Task Architecture Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4822–4829.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021. Call me sexist, but...: Revisiting Sexism Detection Using Psychological Scales and Adversarial Samples. In *Proceedings of the Fifteenth International Conference on Web and Social Media*. AAAI Press.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of NLI models: Uncovering the influence of three factors on robustness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1975–1985, New Orleans, Louisiana. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, NeurIPS'2019.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042, Online. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. How does counterfactually augmented data impact models for social computing constructs? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 325–344, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. dEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 395–405.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR*, abs/1312.6034.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733, Online. Association for Computational Linguistics.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Dominik Stammbach and Elliott Ash. 2020. e-FEVER: Explanations and Summaries for Automated Fact Checking. In *Proceedings of the 2020 Truth and Trust Online Conference (TTO 2020)*, page 32. Hacks Hackers.

Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109, Hong Kong, China. Association for Computational Linguistics.

Ilia Stepin, Jose M. Alonso, Alejandro Catala, and Martín Pereira-Fariña. 2021. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access*, 9:11974–12001.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5594–5608, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. Learning What Makes a Difference from Counterfactual Examples and Gradient Supervision. In *Computer Vision – ECCV 2020*, pages 580–599, Cham. Springer International Publishing.

Paul Thagard. 1989. Explanatory coherence. *Behavioral and brain sciences*, 12(3):435–467.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A Survey on Explainability in Machine Reading Comprehension.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2022. Going Beyond Approximation: Encoding Constraints for Explainable Multi-hop Inference via Differentiable Combinatorial Solvers.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019a. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2944–2953, Hong Kong, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019b. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcos Treviso and André F. T. Martins. 2020. The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118, Online. Association for Computational Linguistics.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing entailment for multi-hop question answering tasks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2948–2958, Minneapolis, Minnesota. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*.

Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1229–1239, Varna, Bulgaria. INCOMA Ltd.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of ICLR*.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Jörg Wagner, Jan Mathias Köhler, Tobias Gindele, Leon Hetzel, Jakob Thaddäus Wiedemer, and Sven Behnke. 2019. Interpretable and Fine-Grained Visual Explanations for Convolutional Neural Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9089–9099. IEEE.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

Ziqi Wang*, Yujia Qin*, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2020. Learning from Explanations with Neural Execution Tree. In *International Conference on Learning Representations*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach Me to Explain: A Review of Datasets for Explainable Natural Language Processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.

Dustin Wright and Isabelle Augenstein. 2020. Claim check-worthiness detection as positive unlabelled learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 476–488, Online. Association for Computational Linguistics.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland. Association for Computational Linguistics.

Lianwei Wu, Yuan Rao, Yongqiang Zhao, Hao Liang, and Ambreen Nazir. 2020. DTCA: Decision tree-based co-attention networks for explainable claim verification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1024–1035, Online. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT Contextual Augmentation. In *Computational Science – ICCS 2019*, pages 84–95, Cham. Springer International Publishing.

Brian Xu, Mitra Mohtarami, and James R. Glass. 2018. Adversarial Domain Adaptation for Stance Detection. In *Proceedings of the Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS)–Continual Learning*.

Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4514–4525, Online. Association for Computational Linguistics.

Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Hu. 2019. XFake: Explainable Fake News Detector with Visualizations. In *The World Wide Web Conference*, pages 3600–3604.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (In)fidelity and Sensitivity of Explanations. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fan Yin, Zhouxing Shi, Cho-Jui Hsieh, and Kai-Wei Chang. 2022. On the sensitivity and stability of model interpretations in NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2631–2647, Dublin, Ireland. Association for Computational Linguistics.

Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium. Association for Computational Linguistics.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL machine reading group: Four factor framework for fact finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102, Brussels, Belgium. Association for Computational Linguistics.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4094–4103, Hong Kong, China. Association for Computational Linguistics.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using “annotator rationales” to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York. Association for Computational Linguistics.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2008. Machine Learning with Annotator Rationales to Reduce Annotation Cost. In *Proceedings of the NIPS*2008 Workshop on Cost Sensitive Learning*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Tal Zarsky. 2016. The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. *Science, Technology, & Human Values*, 41(1):118–132.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020b. Adversarial Attacks on Deep-Learning Models in Natural Language Processing: A Survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

Ye Zhang, Iain Marshall, and Byron C. Wallace. 2016. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 795–804, Austin, Texas. Association for Computational Linguistics.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-XH: Multi-evidence Reasoning with Extra Hop Attention. In *The Eighth International Conference on Learning Representations (ICLR 2020)*.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

Zhi-Hua Zhou and Ming Li. 2005. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Trans. on Knowl. and Data Eng.*, 17(11):1529–1541.