Benchmarking the Generation of Fact Checking Explanations

Daniel Russo[1,2], **Serra Sinem Tekiroğlu**[1], **Marco Guerini**[1]
[1]Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento, Italy
drusso@fbk.eu, tekiroglu@fbk.eu, guerini@fbk.eu
[2]University of Trento, Italy

Abstract

Fighting misinformation is a challenging, yet crucial, task. Despite the growing number of experts being involved in manual fact-checking, this activity is time-consuming and cannot keep up with the ever-increasing amount of fake news produced daily. Hence, automating this process is necessary to help curb misinformation. Thus far, researchers have mainly focused on claim veracity classification. In this paper, instead, we address the generation of justifications (textual explanation of *why* a claim is classified as either true or false) and benchmark it with novel datasets and advanced baselines. In particular, we focus on summarization approaches over unstructured knowledge (i.e., news articles) and we experiment with several extractive and abstractive strategies. We employed two datasets with different styles and structures, in order to assess the generalizability of our findings. Results show that in justification production summarization benefits from the claim information, and, in particular, that a claim-driven extractive step improves abstractive summarization performances. Finally, we show that although cross-dataset experiments suffer from performance degradation, a unique model trained on a combination of the two datasets is able to retain style information in an efficient manner.

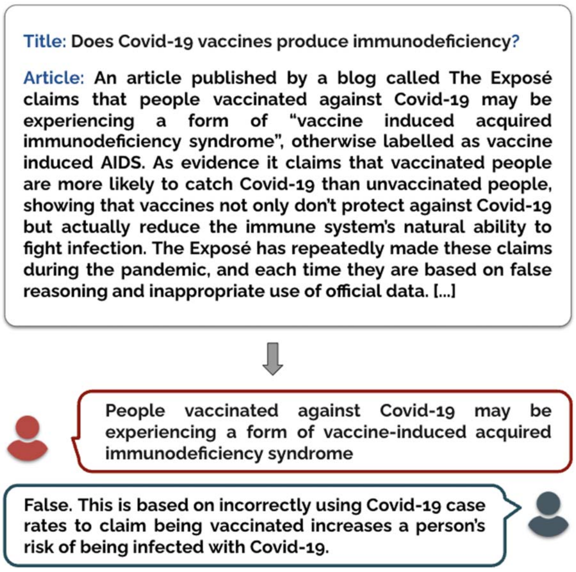
1 Introduction

The interaction between the modern media ecosystem and online social media has facilitated the rapid and nearly unrestricted spreading of news. While this has been a major achievement in terms of access to information, there is also an increasing need to counter the spread of misinformation, commonly conveyed through fake news. Fake news is crafted with the intention to manipulate society towards a specific political, economic, or social outcome, lacking verifiable evidence and credible sources (Chen and Sharma, 2015). It can represent a threat to human health and safety, e.g., by disseminating false information on disease treatment (Van der Linden, 2022). Thus, verifying the accuracy of claims and presenting users with factual and impartial evidence to support their veracity is of utmost importance. Manual fact-checking, however, is a time-consuming activity (Hassan et al., 2015). Hence, Natural Language Processing has been suggested as an effective solution for automating this process. Thus far, the main strategies have involved classifying and flagging misleading information. However, a simple classification approach can generate a *backfire effect* where the belief of false claims is further entrenched rather than hindered (Lewandowsky et al., 2012). For this reason, explaining *why* a claim is classified as either true or false can be a better solution. Fact-checking articles could represent a valuable resource towards this end, however, on online social media platforms they are ineffective either because ordinary users are not prone to click on links to relevant resources (Glenski et al., 2017, 2020) or because these articles are excessively long to the point that users would avoid reading it (Pernice et al., 2019). Indeed, effective explanations should be simple, and only a few arguments must be provided in order to avoid an *''overkill'' backfire effect* (Lombrozo, 2007; Sanna and Schwarz, 2006).

Although the work of professional fact-checkers is crucial for countering misinformation (Wintersieck, 2017), it has been shown that disproof on social media platforms is mostly carried out by ordinary users (Micallef et al., 2020). Thus, automating the explanation generation process is deemed crucial, as an aid for both fact-checkers (to increase their online activity) and for social media users (to make their intervention more effective; He et al., 2023).

Still, few attempts to automatically generate explanations/justifications about claim veracity

Transactions of the Association for Computational Linguistics, vol. 11, pp. 1250–1264, 2023. https://doi.org/10.1162/tacl_a_00601
Action Editor: Roi Reichart. Submission batch: 1/2023; Revision batch: 6/2023; Published 10/2023.



Title: Does Covid-19 vaccines produce immunodeficiency?

Article: An article published by a blog called The Exposé claims that people vaccinated against Covid-19 may be experiencing a form of "vaccine induced acquired immunodeficiency syndrome", otherwise labelled as vaccine induced AIDS. As evidence it claims that vaccinated people are more likely to catch Covid-19 than unvaccinated people, showing that vaccines not only don't protect against Covid-19 but actually reduce the immune system's natural ability to fight infection. The Exposé has repeatedly made these claims during the pandemic, and each time they are based on false reasoning and inappropriate use of official data. [...]

People vaccinated against Covid-19 may be experiencing a form of vaccine-induced acquired immunodeficiency syndrome

False. This is based on incorrectly using Covid-19 case rates to claim being vaccinated increases a person's risk of being infected with Covid-19.

Figure 1: An article (top) used to generate a verdict (bottom) in response to a false claim (middle).

have been proposed so far (Kotonya and Toni, 2020a). Current methods for justification production include highlighting tokens with high attention weights (Popat et al., 2018; Yang et al., 2019; Lu and Li, 2020), utilizing knowledge graphs (Ahmadi et al., 2019), and modeling it as either an extractive or abstractive summarization task (Atanasova et al., 2020; Kotonya and Toni, 2020b).

In this paper, we aim at benchmarking justification production as a summarization task, by providing an exhaustive study of the performance of extractive and abstractive approaches over two novel datasets. In particular, we consider several extractive and abstractive approaches both in supervised and unsupervised settings, where we generate a justification for a given claim using a fact-checking article as a knowledge source. We also experiment with hybrid approaches combining extractive and abstractive steps in a unique pipeline. Finally, we integrate the pipeline within an end-to-end claim-driven explanation generation framework. These approaches are tested both in in-domain and cross-domain configurations, by employing two different datasets. Each dataset has its own style and characteristics, but they both contain claim, verdict, and article triplets (see Figure 1).

The main findings from our experiments are: (i) If an extractive approach is employed for justification production, then the sentence selection must be driven by the claim information. (ii) If no training data is available in cross-domain experiments, extractive approaches can be better than abstractive ones for justification production. (iii) High-quality justifications can be obtained by combining in a unique pipeline extractive and abstractive summarization approaches (using simple off-the-shelf language models [LMs]), and by driving sentence selection and justification generation with the claim information. Still, differently from previous studies, we found that the sentences extracted from the article must retain their order rather than being rearranged according to some notion of relevance. (iv) LMs for abstractive summarization should be selected according to article length since there is not a one-fits-all solution: For shorter articles, 512 tokens input length LMs provide better results, while using models with 1024 input length is beneficial for longer examples. (v) Although cross-dataset experiments suffer from performance degradation, LM-based models are able to retain different verdict styles: Fine-tuning a single LM on the union of datasets with different stylistic characteristics leads to performance similar to those obtained by fine-tuning a model for every single dataset.

2 Related Work

The process of fact-checking a news story involves determining the truthfulness of a statement (*Verdict Prediction*) and the generation of a written rationale for the verdict (*Justification Production*). The claim veracity is usually assessed through a classification task, both binary (Nakashole and Mitchell, 2014; Potthast et al., 2018; Popat et al., 2018) and multi-class (Wang, 2017; Thorne et al., 2018), or through a multi-task learning approach (Augenstein et al., 2019). Recently, researchers are focusing on developing datasets and systems for evidence-based Verdict Prediction. Among the most relevant datasets, notable examples include the FEVER dataset (Thorne et al., 2018), SciFact (Wadden et al., 2020), COVID-fact (Saakyan et al., 2021), and PolitiHop (Ostrowski et al., 2021).

Justification Production has proven to be more challenging than *Verdict Prediction*. Several approaches have been suggested, including logic-based approaches (Gad-Elrab et al., 2019; Ahmadi et al., 2019) or deep-learning and attention-based techniques (Popat et al., 2018; Yang et al., 2019; Shu et al., 2019; Lu and Li, 2020). Nevertheless, casting justification production as a summarization task appears to be the most viable solution (Kotonya and Toni, 2020a). Thereby, explanations can be derived from manually written debunking articles either by selecting important sentences from the text (*extractive approach*; Atanasova et al., 2020) or by generating a new one (*abstractive approach*; Kotonya and Toni, 2020b). Extractive and abstractive summarization approaches still have many problems: Extractive-generated explanations cannot generate sufficiently context-full explanations, while abstractive-generated ones may lack faithfulness, given the tendency to hallucinate of these neural models (Kotonya and Toni, 2020a; Guo et al., 2022). Currently, the abstractive summarization technique appears to be the most viable option for generating effective justifications. Nevertheless, it may not always be possible to acquire an adequate amount of training data or the necessary computational resources for highly demanding models. Thus, the purpose of this paper is twofold: (i) provide SOTA results using simple off-the-shelf LMs, and (ii) understand which is the most suitable approach for a given scenario.

3 Datasets

For our experiments, we collected two datasets with different structural and stylistic features. The first is LIAR++, a derivation of LIAR-PLUS (Alhindi et al., 2018), and the second is FullFact, a completely new dataset. Both datasets comprise claim, verdict, and article entries.

The *claim* is a short text consisting of a statement that is under inspection: it can be TRUE, partially TRUE, or FALSE. The *verdict* is usually a paragraph-long text that provides arguments to assess the truth value of the claim: In many cases, it corresponds to a debunking text.[1] Finally, the *article* is a document that discusses the veracity of the claim using a journalistic style and contains the verifiable facts necessary to build the verdict. Figure 1 illustrates an example for each element. A detailed description of the employed datasets follows.[2]

3.1 LIAR++ Dataset

We created LIAR++ (L_{++} henceforth) starting from the LIAR-PLUS dataset (Alhindi et al., 2018). This dataset contains articles from the POLITIFACT website[3] spanning from 2007 to 2016 and covers various political topics with a primary emphasis on verifying the accuracy of statements made by political figures. LIAR-PLUS contains some entries in which the verdict was *artificially created* by extracting the last five sentences from the body of the article. In all the other cases, verdicts were extracted from a specific section of web pages, usually titled *Our ruling* or *Summing up*. Qualitative and quantitative analyses of the *artificial* against *gold* verdicts showed that the former did not meet the expected quality. Therefore we decided to discard them while creating L_{++}. Differently from LIAR-PLUS, we also kept the whole verdict without removing the 'forbidden sentences' (i.e., sentences comprising any verdict-related word) such as ``*this statement is false*''.[4] After this procedure L_{++} comprises 6451 *claim-article-verdict* triples.

3.2 FullFact Dataset

With a similar procedure to that used for L_{++}, we created a new dataset starting from the FULLFACT website[5] (FF henceforth). This dataset contains data spanning from 2010 to 2021, and covers several different topics, such as health, economy, crime, law, and education. In FF the verdict is always present as a separate element in the web page so there was no need to filter the data. This dataset accounts for 1838 *claim-article-verdict* triples.

3.3 Analysis of the Datasets

In this section, we focus on the main structural and stylistic differences between the two datasets, especially those that can have an impact on the experiments presented in the following sections. We mainly employed ROUGE score (Lin, 2004) as evaluation metric in order to assess the quality of our datasets and of the generated summaries. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) counts the number of overlapping units between two different texts. In the

[1] In the literature, the term verdict often indicates the degree of truthfulness of a claim, and it is usually expressed as a label. Instead, our verdict contains also the so-called *justification* or *explanation* of the verdict label.

[2] The code for dataset creation can be found at the following link `https://github.com/LanD-FBK/benchmark-gen-explanations`.

[3] `https://www.politifact.com`.

[4] LIAR-PLUS was meant for claim classification, thereby those forbidden sentences would have made the task trivial.

[5] `https://fullfact.org`.

		SENT$_\mu$	TOK$_\mu$	BPE$_\mu$
L$_{++}$	Article	38.9	817.8	1131.7
	Claim	1.2	17.9	24.9
	Verdict	6.3	113.7	150.4
FF	Article	24.8	632.1	803.5
	Claim	1.0	15.0	20.3
	Verdict	1.9	30.4	39.0

Table 1: Average length of each element of the datasets in terms of number of sentences (SENT), standard tokens (TOK), and BPE tokens (BPE).

paper we report: ROUGE-N (*R-N*, N=1,2) which counts the number of *n*-grams overlapping, and ROUGE-L (*R-L*) taking into account the lowest common subsequence between two texts.

Average Article and Verdict Length. Data length was computed in terms of the number of sentences, standard tokens, and byte-pair encoding (BPE) tokens.[6] As shown in Table 1, FF articles and verdicts are much shorter than the L$_{++}$ counterparts: 632 vs. 818 tokens and 30 vs. 114 tokens, respectively. On the contrary, claim lengths are essentially similar (18 vs. 15). Regarding the lengths in terms of BPE tokens, the average length of articles alone exceeds the fixed input length of the major LMs, which is usually 512 or 1024 (see Table 1). Indeed, 98% and 54% of L$_{++}$ articles are above the 512 and 1024 limit, respectively, while these are 66% and 24%, respectively, for FF. This implies that input reduction or truncation will be needed when processing the data during our experiments.

Presence of Verdict Snippets in the Article. We compared the two datasets in terms of the possibility of abstracting/extracting the verdict from the article. In particular, we considered ROUGE recall to highlight how many verdict snippets are present in the article. Results indicate that L$_{++}$ has a more abstractive nature than FF (see Table 2). Indeed, the text of the verdict is present in the article more verbatim for FF than for L$_{++}$ (0.547 vs. 0.426 ROUGE-L recall). On the contrary, with ROUGE F1 we can observe how difficult it is to find verdict material in the article. Results show that FF articles contain very few pieces of FF verdicts. This can be explained in light of the much shorter length of the FF verdicts as compared to L$_{++}$ ones (39 vs. 150 BPE tokens on average, see Table 1), while article length difference is negligible in this comparison.

		R1	R2	RL
L$_{++}$	Rec.	0.678	0.272	0.426
	F1	0.168	0.067	0.103
FF	Rec.	0.724	0.355	0.547
	F1	0.093	0.045	0.068

Table 2: Verdict and article overlap measured in terms of ROUGE F1 and Recall scores.

To sum up, FF verdicts are much shorter than L$_{++}$ verdicts and even if they are present in longer verbatim sequences in the articles, these sequences are much more spread out the document. Thus, we expect that it will be harder to identify and extract FF verdicts.

Claim Repetition in Verdict. The possible presence of significant parts of the claim in the verdict positively affects the ROUGE scores without necessarily indicating a better verdict quality.[7] For example, a trivial baseline that, given a claim, outputs a verdict that simply states ''*It is not true that [claim]*'' would obtain a high ROUGE score without producing any significant explanation to a verdict. Thus, we analyzed claim and verdict overlap and report the results in Table 3. Considering ROUGE-L, on average 65% of claims' subsequences are quoted verbatim in the verdict for the L$_{++}$ dataset, while only 26% for FF. The frequent reference to the claim at the beginning of the verdict can explain this outcome (Example in Appendix A, Table 13). To check this hypothesis we re-computed ROUGE scores after removing the first sentence of the claim. We also repeated the test by removing the last sentence as a control condition. We observe that for L$_{++}$ ROUGE scores drop when evaluating the overlap between claim and verdict without the first sentence (i.e., ROUGE-1 goes from 0.709 to 0.394). On the other hand, this is not the case with the

[6] Computed using T5-large tokenizer.

[7] For example, ''*The statement that [People vaccinated against Covid-19 may acquire immunodeficiency syndrome] was originally posted by ...*''

	Verdict	R1	R2	RL
L_{++}	comp.	0.709	0.532	0.648
	no 1st	0.394	0.130	0.302
	no last	0.702	0.527	0.643
FF	comp.	0.311	0.099	0.257
	no 1st	0.247	0.074	0.208
	no last	0.192	0.061	0.165

Table 3: Recall of ROUGE scores between claims and verdicts. *comp.* indicates scores compute on the whole verdict, while *no 1st* and *no last* indicates the removal of the first and last sentence, respectively.

	Article	R1	R2	RL
L_{++}	comp.	0.875	0.706	0.810
	no 1st	0.862	0.689	0.791
	no last	0.874	0.704	0.808
FF	comp.	0.785	0.426	0.661
	no 1st	0.739	0.351	0.612
	no last	0.779	0.421	0.655

Table 4: Recall of ROUGE scores between claims and articles. *comp.* indicates scores compute on the whole verdict, while *no 1st* and *no last* indicates the removal of the first and last sentence, respectively.

removal of the last sentence (R1 is 0.702), which corroborates our hypothesis.

To sum up, L_{++} verdicts include a good amount of claim information, usually reported in the first sentence. However, this does not apply to FF. Additionally, L_{++} verdicts end with a statement about claim veracity, usually in the form *“We rate the claim `[TRUTH LABEL]`”*.

Article Adherence to the Claim. The amount of text of the claim included in the article is a proxy for understanding: (i) if a simple summarization approach could provide a good verdict, even without explicitly providing the claim, and (ii) if there is a preferable portion of the article to be selected for summarization in order to fit into LMs' input length.

Since we know that the articles are written to discuss the veracity of the corresponding claims, we expect each article to contain a certain amount of information related to the claim, including partial or even whole quotations of it. This assumption would be reflected in high ROUGE recall values between the claim and the article.

Results in Table 4 confirm our expectations. While the high ROUGE-1 values can be trivially explained by the claim and article having the same topic, the high ROUGE-2 and L recall values indicate that entire portions of the claim were inserted into the article. On average, 80% of the claim subsequences are quoted verbatim within the article in the two datasets. However, verbatim claim text is not particularly used in the first sentence of the article: Its content is spread over the article, as can be seen by the small variation in ROUGE scores obtained by removing the first or last sentences of each document.

To sum up, the claim information is highly present within the article and spread over the entire text. For this reason, we expect extractive summarization approaches to be better than simple text truncation at selecting meaningful information from the article.

4 Experimental Setup

In this section, we present several experiments for the task of justification production. All the approaches can be traced back to the pipeline presented in Figure 2. Given an article, we tested several extractive approaches to select relevant material. Extractive summarizes were considered Justifications per se or were sent to a LM pre-trained on the abstractive summarization objective. The LMs, in turn, were used with or without a fine-tuning step. Moreover, we selected different decoding mechanisms to drive the generation. Eventually, we conduct a cross-domain experiment to evaluate the models' robustness to the style of each dataset.

4.1 Extractive Approaches

We first explored unsupervised extractive methods by comparing three different settings: article truncation, article-relevance extractive summarization (using LexRank algorithm), and claim-driven extractive summarization (with SBERT). Each configuration represents a different assumption: (i) the main content (corresponding to a possible verdict) is introduced at the beginning or at the end of the article, within a specific section; (ii)

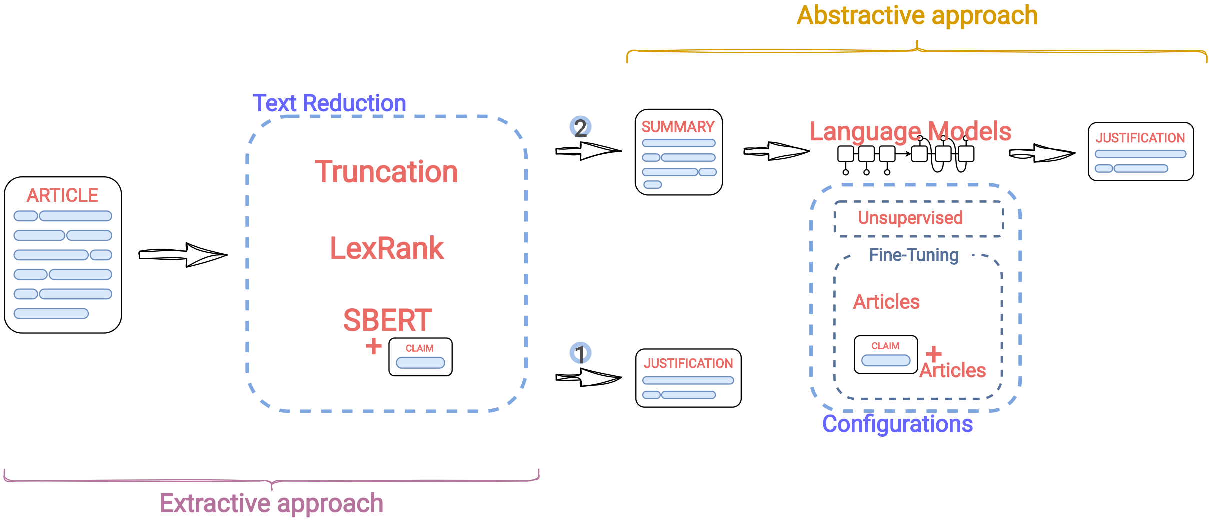


Figure 2: General pipeline of our experiments for Justification Production. (1) extractive approach only; (2) extractive and abstractive summarization approaches combined in a unique pipeline. The total number of 960 configurations/experiments comprises 2 datasets (FF and L_{++}) × 10 summary configurations (8 from LexRank and SBERT both top/bottom and article/ranking order + 2 from Truncation head/tail) × 4 LMs (T5, dBart, Peg_{xsum}, Peg_{cnn}) × 3 fine-tuning (`unsupervised`, `article`, `claim+article`) × 4 decodings (beam search, Top-K sampling, nucleus sampling, and typical sampling).

a proper extractive summary or verdict contains the most relevant sentences of the article; (iii) a proper verdict comprises the article sentences most similar to the claim.

- **Truncation** is the most straightforward approach of ‘‘input reduction’’, i.e., cutting the input at a given threshold. This is the simplest procedure applied when using LMs on long texts.
- **LexRank** (Erkan and Radev, 2004) is an unsupervised approach for extractive text summarization which ranks the sentences of a document through a graph-based centrality scoring.
- **SBERT** (Reimers and Gurevych, 2019) is a Siamese network based on BERT (Devlin et al., 2019) employed for generating and ranking sentence embeddings with respect to a target sentence (i.e., the claim) using cosine-similarity.

All the reduction baselines were tested under two configurations: From the list of sentences they provide, we selected either the top or bottom of the list. Furthermore, for LexRank and SBERT we rearranged top or bottom sentences according to article or ranking order.

4.2 Abstractive Approach

In the second part of our experimental design, we combined extractive and abstractive summarization for justification production. A reduced version of the text, obtained through truncation or extractive summarization, was used as input to various off-the-shelf Transformer-based models pre-trained on an abstractive summarization objective. In particular, we experiment with 4 Transformer-based summarization LMs[8] trained on news-specific summarization datasets:

- **T5**: T5-large, 738M parameters, input size 512, (Raffel et al., 2020)
- **Peg_{xsum}**: Pegasus xsum, 570M parameters, input size 512 (Zhang et al., 2020)
- **Peg_{cnn}**: Pegasus cnn_dailymail, 570M parameters, input size 1024 (Zhang et al., 2020)
- **dBart**: DistilBart cnn-12-6, 305M parameters, input size 1024 (Shleifer and Rush, 2020)

All the models were tested under three main configurations: `unsupervised`, fine-tuned on

[8] We have used the Huggingface Transformers library for our experiments: `https://huggingface.co/transformers`.

	Method	R1	R2	RL
FF	truncation	0.258	0.082	0.182
	LexRank	0.267	0.083	0.180
	SBERT	**0.300**	**0.114**	**0.213**
L_{++}	truncation	0.347	0.120	0.196
	LexRank	0.373	0.120	0.194
	SBERT	**0.393**	**0.158**	**0.237**

Table 5: Extractive approaches comparison. The number of sentences to be extracted is set to the average number of sentences per verdict in the corresponding datasets (2 for FF and 6 for L_{++}).

a reduced version of the article (`article`), and fine-tuned on the concatenation of the claim and the reduced article (`claim+article`). Finally, four decoding mechanisms were employed for generating the verdicts: *beam search* (5 beams), *Top-K sampling* (sampling pool limited to 40 words), *nucleus sampling* (probability set to 0.9), and *typical sampling* (probability set to 0.95; Meister et al., 2023). The fine-tuning details and hyperparameter settings can be found in Appendix B.

5 Experimental Results

Our experimental design combines all the settings described in the previous sections. Extractive and abstractive approaches are concatenated in a unique pipeline tested on both L_{++} and FF. Additionally, we tested the generalization capabilities of the pipeline in zero-shot experiments and by integrating the two datasets into a unique model. Although we tested the complete design (960 configurations, as depicted in Figure 2), we will discuss only the most relevant findings hereafter.

Claim-driven Extractive Summarization. If we focus on verdict generation as a pure unsupervised extractive summarization task, then *the claim-driven approach through SBERT leads to better results in both datasets* (see Table 5). The second best approach is LexRank, which focuses on sentence relevance within the article (rather than claim relevance). Simple truncation led to the lowest results when considering ROUGE-1. In Table 5 we report the best results, i.e., top selection with article order. Results for bottom selection and ranking order are reported in Table 14 in Appendix C.[9]

	Model	Order	R1	R2	RL
L_{++}	T5	art.	0.448	0.240	0.349
		rank.	0.454	0.242	0.351
	Peg_{xsum}	art.	0.452	0.247	0.355
		rank.	0.455	0.249	0.357
	dBart	art.	0.460	0.254	0.359
		rank.	0.454	0.246	0.352
	Peg_{cnn}	art.	**0.476**	**0.261**	**0.371**
		rank.	0.467	0.255	0.366
FF	T5	art.	0.360	0.139	0.269
		rank.	0.342	0.128	0.257
	Peg_{xsum}	art.	**0.359**	**0.144**	**0.269**
		rank.	0.334	0.121	0.246
	dBart	art.	0.350	0.131	0.255
		rank.	0.358	0.143	0.265
	Peg_{cnn}	art.	0.355	0.138	0.261
		rank.	0.335	0.126	0.248

Table 6: ROUGE F1 scores for each model in the SBERT top `claim+article` configuration. Verdicts were generated through the beam search decoding method (the best among the 4 decoding mechanisms tested). The input length size for T5 and Peg_{xsum} is 512, while for dBart and Peg_{cnn} it is 1024.

Sentence Order for LM Input. An aspect that can have a significant impact on LMs' performance is the order of the sentences fed to the LMs. Results show that *rearranging sentences according to ranking order, rather than article order, can hinder text coherence*. As can be seen in Table 6, article order is generally better than ranking order for 1024 input size LMs with L_{++}. For FF, article order leads to higher ROUGE scores with 512 input-size LMs. Differences among datasets can be explained by the lengths of their articles: in particular, most of the articles from FF are shorter than 512 BPE tokens.

[9] Bottom approaches represent specific assumptions: (i) for truncation, the hypothesis is that informative content is in the last lines of the articles (in the form of a ‘‘to sum up’’ paragraph); (ii) for SBERT that the most similar sentences could be those that simply rephrase the claim but are not necessarily the most informative.

Configuration			R1	R2	RL
L$_{++}$	T5	unsup.	0.293	0.103	0.194
		art.	0.437	0.217	0.328
		cl+art.	**0.448**	**0.240**	**0.349**
	Peg$_{xsum}$	unsup.	0.206	0.056	0.138
		art.	0.442	0.230	0.339
		cl+art.	**0.452**	**0.247**	**0.355**
	dBart	unsup.	0.336	0.115	0.214
		art.	0.452	0.225	0.333
		cl+art.	**0.460**	**0.254**	**0.359**
	Peg$_{cnn}$	unsup.	0.267	0.097	0.189
		art.	0.463	0.238	0.350
		cl+art.	**0.476**	**0.261**	**0.371**
FF	T5	unsup.	0.302	0.103	0.203
		art.	0.331	0.117	0.239
		cl+art.	**0.360**	**0.139**	**0.269**
	Peg$_{xsum}$	unsup.	0.241	0.061	0.174
		art.	0.329	0.125	0.244
		cl+art.	**0.359**	**0.144**	**0.269**
	dBart	unsup.	0.284	0.100	0.187
		art.	0.320	0.113	0.233
		cl+art.	**0.350**	**0.131**	**0.255**
	Peg$_{cnn}$	unsup.	0.281	0.094	0.196
		art.	0.319	0.113	0.234
		cl+art.	**0.355**	**0.138**	**0.261**

Table 7: ROUGE F1 scores for each model in the SBERT top configuration. Results for both the unsupervised and the two fine-tuning settings (`article` and `claim+article`) are reported. Verdicts were generated through the beam search decoding method.

		Length	R1	R2	RL
L$_{++}$	unsup.	512	0.249	**0.121**	0.166
		1024	**0.302**	0.106	**0.202**
	art.	512	0.440	0.224	0.334
		1024	**0.458**	**0.232**	**0.342**
	cl+art.	512	0.450	0.244	0.352
		1024	**0.468**	**0.257**	**0.365**
FF	unsup.	512	0.272	0.082	0.189
		1024	**0.283**	**0.097**	**0.192**
	art.	512	**0.330**	**0.121**	**0.242**
		1024	0.320	0.113	0.234
	cl+art.	512	**0.360**	**0.142**	**0.269**
		1024	0.353	0.135	0.258

Table 8: Averaged results for models' input length. ROUGE scores for verdicts generated under the SBERT, article order, top, `claim+article` configuration (beam search decoding).

Claim-driven Abstractive Summarization. One major question when using LMs is whether the claim information is essential to drive the generation of the justification. Indeed, we should consider that (i) the sentences used as LM input are already selected according to the claim (SBERT) (ii) we are using gold articles (i.e., specifically written to debunk the given claim). *Results show that the enrichment of the input with the claim information leads to ROUGE scores even higher than those obtained through a simple fine-tuning on the articles only* (see Table 7).

LM Input Length. Throughout the experiments, we saw that 1024 input-length models had higher results on L$_{++}$, while on FF better performance was recorded with 512 input-length models (see Table 8). A possible explanation is that the differences in performance are due to the average length of articles in the two datasets (longer for L++, shorter for FF, see Table 1). In order to provide additional evidence for this hypothesis, we calculated ROUGE scores exclusively for articles with a length of 512 BPE tokens or less from both datasets. The results indicate that ROUGE scores for 512 input models were higher in both datasets than those obtained with 1024, *proving that article length is the key factor when selecting the proper model.*

Extractive vs Abstractive Summarization. In most cases, extractive summarization is better than unsupervised abstractive summarization (especially when claim-driven) in terms of ROUGE scores. Thus, *if no training data is available, claim-driven extractive summarization is a viable solution*. On the other hand, *when training data is available the best approach is to combine claim-driven abstractive and extractive summarization.*

6 Cross-data and Mixed-data Experiments

Next, we explored the impact of the datasets' stylistic characteristics in several training/test configurations. First, we conducted a zero-shot cross-dataset experiment, then we investigated the effect of combining the two datasets for training a single model.

`claim` : There have been 1,400 deaths and one million injured from Covid-19 vaccinations in the UK.

`gold verdict` : These are deaths and potential side effects reported following the vaccine, not necessarily because of it.

`SBERT` : The front page of free newspaper ‘The Light’, shared on Facebook, claimed that there have been 1,400 deaths and a million injuries ‘‘from covid injections’’ in the UK. There had been just over 1,470 deaths following a Covid-19 vaccination in the UK, according to the Medicines and Healthcare products Regulatory Agency’s (MHRA) Yellow Card reporting scheme, as of 7 July 2021, when the paper came out.

`abstractive` : This is technically correct, but the fact that a death is reported following a vaccination is no proof the vaccine was the cause of this death or injury.

Table 9: FF example of generated verdicts. The first one is generated through extractive summarization only (with SBERT); the second example is the output of the extractive and abstractive pipeline (SBERT, top, article order, `claim+article` configuration with Peg$_{cnn}$).

		R1	R2	RL
L_{++} →FF articles	T5	0.274	0.087	0.181
	Peg$_{xsum}$	0.277	0.100	0.195
	Peg$_{cnn}$	0.266	0.089	0.182
	dBart	0.266	0.093	0.174
L_{++} →FF claim+art	T5	0.288	0.092	0.194
	Peg$_{xsum}$	0.282	0.109	0.200
	Peg$_{cnn}$	0.286	0.105	0.198
	dBart	0.278	0.098	0.191
FF→L_{++} articles	T5	0.256	0.087	0.171
	Peg$_{xsum}$	0.248	0.084	0.164
	Peg$_{cnn}$	0.245	0.084	0.167
	dBart	0.269	0.081	0.172
FF→L_{++} claim+art.	T5	0.271	0.099	0.184
	Peg$_{xsum}$	0.262	0.099	0.178
	Peg$_{cnn}$	0.244	0.087	0.168
	dBart	0.274	0.092	0.180

Table 10: Models fine-tuned on FF and tested on Liar+ test set (FF→L_{++}) and fine-tuned on L_{++} and tested on FF test set (L_{++} →FF), using `article` or `claim+article` configurations (SBERT top article order).

Cross-dataset Experiments. In these experiments, the models were fine-tuned on one dataset and tested on the other, i.e., fine-tuned on L_{++} and tested on FF (L_{++}→FF) and vice-versa (FF→L_{++}) both for `article` and `claim+article` configurations. In particular, we employed the best-performing pipeline from the previous experiments, i.e., models from the SBERT, top, article order configuration. Results are reported in Table 10. Both for L_{++} and FF, the models show the trend highlighted previously: The `claim+article` configuration performs better than the `article` configuration. Furthermore, as expected, testing on a different dataset yielded lower results: In several cases, results for the `article` configuration were on par or even worse than those obtained with the unsupervised LMs (compare with Table 7). This is particularly evident for the FF→L_{++} configuration. The low ROUGE values can be attributed to the distinct styles of the datasets and not to any degradation in the generation quality. As can be seen from the examples in Table 11, models fine-tuned on L_{++}, even when tested on FF, generate justifications mimicking L_{++} style (claim in the first sentence and truthfulness statement at the end, see Appendix A), and vice versa (FF→L_{++}).

Mixed Data Experiments. Finally, we tested the effect of using both datasets in training a single LM. We focused on Peg$_{cnn}$ as in the in-domain experiments it generally showed quantitatively (see Table 7) and qualitatively (see Table 9) better results, especially for longer input sequences. To this end, we combined the training data in a unique unbalanced dataset and we tested the extractive and abstractive pipeline (SBERT, top, article order setting). Results, reported in Table 12, were found to be comparable to those achieved through the in-domain fine-tuning of distinct models for each dataset (see Table 7). Thus, if the datasets have peculiar styles, a more efficient way to tackle the task is to fine-tune a unique LM on all the data available rather than fine-tuning different models for each dataset.

7 Conclusions

Curbing misinformation with NLP tools is a crucial task. Up to now, researchers have mainly

L_{++} →FF
`claim`: University applications have fallen by 4%. `gold verdict`: The number of applicants through UCAS to UK higher education institutions has gone down by 4% compared to last year. `generated verdict`: The total number of people applying to higher education in the UK this year (as of the final June 2017 deadline) is down 4% compared to last year, driven by falls from England, Wales and the EU. There have been a number of reasons for the overall drop, including a fall in the number of 18 year olds in the UK population. We rate this claim True.
FF→L_{++}
`claim`: Muslim nations did not call out the people who celebrated the 9/11 attacks. `gold verdict`: O'Reilly said that Muslim nations did not ''call out'' the people who celebrated the 9/11 attacks. So far as we can tell, there was no official condemnation of people celebrating the 9/11 attacks. However, Muslim governments, and religious leaders, condemned the attacks themselves, as did many average Muslims. O'Reilly's statement is accurate as far as it goes, but it leaves out important information that might lead someone to reach a different conclusion. That meets our definition of Half True. `generated verdict`: There was no official condemnation from a Muslim-majority nation. What we did find were many official condemnations of the attacks themselves. Average Muslims attended candlelight vigils and other public events to voice sympathy for the victims and to repudiate the attacks.

Table 11: Examples from the cross-data experiments.

	R1	R2	RL
L_{++}	0.473	0.261	0.370
FF	0.367	0.143	0.272

Table 12: F1 ROUGE scores of Peg_{cnn} fine-tuned on a unique dataset and tested on L_{++} and FF test sets.

focused on claim veracity classification. In this paper, instead, we focused on generating textual justifications with factual and objective information to support a verdict. We started casting the problem as a news article summarization task and subsequently we integrated summarization within an end-to-end claim-driven explanation generation framework, accounting for the several practical scenarios that can be encountered. To this end, we experimented with several extractive and abstractive approaches, leveraging pre-trained LMs under manifold configurations. In order to provide an exhaustive benchmark of the justification production task, we employed two novel datasets throughout the experiments. The main results show that summarization needs to be driven by the claim to obtain better performances and that an extractive step before LM abstractive summarization further improves the results. Finally, we show that style information can be retained by a single model which is able to handle multiple datasets at once.

Limitations

LMs suffer from hallucination (Zellers et al., 2019; Solaiman et al., 2019) and, even if the phenomenon is reduced by the document-driven nature of the task, it is still present. In particular, some hallucinations are critical: We occasionally obtain the sentence ''*we rate this statement as false*'' even if the statement is true since it is a very common sentence in the L_{++} training set.

Moreover, the datasets used for this task (i) are restricted to the English language and (ii) assume that there is always a gold article for fact-checking. In real scenarios we might have the debunking material spread over several articles: In this case, we can expect that models not suffering from the input size limit would be most beneficial. Still, from preliminary experiments, we conducted with two long input LMs on our datasets, namely, LED-Large (Beltagy et al., 2020) and BertSumExtAbs (Liu and Lapata, 2019) from Kotonya and Toni (2020b), results were worse also for articles exceeding the 1024 input limit.

Another aspect that should be addressed is an in-depth analysis of automatically generated verdicts and their persuasiveness. In fact, different versions of a verdict for the same claim can have different effects depending on the audience—e.g., for some people explanations comprising few arguments are more effective than longer explanations (Sanna and Schwarz, 2006). To this end, carefully designed human evaluation experiments are needed.

Acknowledgments

We would like to thank our TACL Action Editor and the three anonymous reviewers for their constructive feedback during the review process. This work was partly supported by the AI4TRUST project - AI-based-technologies for trustworthy solutions against disinformation (ID: 101070190).

References

Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. *ArXiv*, cs.DB/1906.09198. Version 1. https://doi.org/10.36370/tto.2019.15

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. https://doi.org/10.18653/v1/W18-5513

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. 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. https://doi.org/10.18653/v1/2020.acl-main.656

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. https://doi.org/10.18653/v1/D19-1475

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv*, cs.CL/2004.05150. Version 2.

Rui Chen and Sushil K. Sharma. 2015. Learning and self-disclosure behavior on social networking sites: The case of facebook users. *European Journal of Information Systems*, 24:93–106. https://doi.org/10.1057/ejis.2013.31

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.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479. https://doi.org/10.1613/jair.1523

Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 87–95, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3289600.3290996

Maria Glenski, Corey Pennycuff, and Tim Weninger. 2017. Consumers and curators: Browsing and voting patterns on reddit. *IEEE Transactions on Computational Social Systems*, 4(4):196–206. https://doi.org/10.1109/TCSS.2017.2742242

Maria Glenski, Svitlana Volkova, and Srijan Kumar. 2020. *User Engagement with Digital Deception*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-42699-6_3

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206. https://doi.org/10.1162/tacl_a_00454

Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The quest to automate fact-checking. In *Proceedings of the 2015 computation+ journalism symposium*. Citeseer.

Bing He, Mustaque Ahamad, and Srijan Kumar. 2023. Reinforcement learning-based counter-misinformation response generation: A case study of covid-19 vaccine misinformation. In *Proceedings of the ACM Web Conference 2023* WWW '23, pages 2698–2709, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3543507.3583388

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. https://doi.org/10.18653/v1/2020.coling-main.474

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. https://doi.org/10.18653/v1/2020.emnlp-main.623

Stephan Lewandowsky, Ullrich K. H. Ecker, Colleen M. Seifert, Norbert Schwarz, and John Cook. 2012. Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest*, 13(3):106–131. https://doi.org/10.1177/1529100612451018, PubMed: 26173286

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sander Van der Linden. 2022. Misinformation: Susceptibility, spread, and interventions to immunize the public. *Nature Medicine*, 28(3):460–467. https://doi.org/10.1038/s41591-022-01713-6, PubMed: 35273402

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. https://doi.org/10.18653/v1/D19-1387

Tania Lombrozo. 2007. Simplicity and probability in causal explanation. *Cognitive Psychology*, 55(3):232–257. https://doi.org/10.1016/j.cogpsych.2006.09.006, PubMed: 17097080

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.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121. https://doi.org/10.1162/tacl_a_00536

Nicholas Micallef, Bing He, Srijan Kumar, Mustaque Ahamad, and Nasir D. Memon. 2020. The role of the crowd in countering misinformation: A case study of the COVID-19 infodemic. *arXiv*, cs.SI/2011.05773. Version 2. https://doi.org/10.1109/BigData50022.2020.9377956

Ndapandula Nakashole and Tom M. Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019, Baltimore, Maryland. Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1095

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. https://doi.org/10.24963/ijcai.2021/536

Kara Pernice, Kathryn Whitenton, and Jakob Nielsen. 2019. *How People Read Online: The Eyetracking Evidence*, 2nd edition. Nielsen Norman Group.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1003

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1022

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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. https://doi.org/10.18653/v1/D19-1410

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. 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 2116–2129, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.165

Lawrence J. Sanna and Norbert Schwarz. 2006. Metacognitive experiences and human judgment: The case of hindsight bias and its debiasing. *Current Directions in Psychological Science*, 15(4):172–176. https://doi.org/10.1111/j.1467-8721.2006.00430.x

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. *arXiv*, cs.CL/2010.13002. Version 2.

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*, KDD '19, pages 395–405, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3292500.3330935

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. Version 2.

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. https://doi.org/10.18653/v1/N18-1074

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.609

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. https://doi.org/10.18653/v1/P17-2067

Amanda L. Wintersieck. 2017. Debating the truth: The impact of fact-checking during electoral debates. *American Politics Research*, 45(2):304–331. https://doi.org/10.1177/1532673X16686555

Fan Yang, Shiva K. Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D. Ragan, Shuiwang Ji, and Xia (Ben) Hu. 2019. Xfake: Explainable fake news detector with visualizations. In *The World Wide Web Conference*, WWW '19, pages 3600–3604, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3308558.3314119

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

A Verdict Stylistic Features

Differently from FF, L_{++} verdicts show a peculiar and recurrent style: The first sentence includes a reference to the claim, usually quoted verbatim (see Table 13). Moreover, verdicts end with a statement about the degree of truthfulness of the related claim, in a form similar to *''We rate the claim `[TRUTH LABEL]`''*. The main justifications are presented in the body of the verdict. Examples are provided in Table 13.

`claim` : Clinton says ''Hate crimes against American Muslims and mosques have tripled after Paris and San Bernardino.''

`verdict` : Clinton said that ''**hate crimes against American Muslims and mosques have tripled after Paris and San Bernardino''.** Calculations by the director of an academic center found that the number did triple after those attacks. But it's worth noting that his data does not show whether or not they remained at that elevated level, or for how long – something that would be a reasonable interpretation of what Clinton said. The statement is accurate but needs clarification or additional information, so **we rate it Mostly True**.

`claim` : Trump says ''I released the most extensive financial review of anybody in the history of politics. ...You don't learn much in a tax return.''

`verdict` : Trump said that he has **''released the most extensive financial review of anybody in the history of politics. . . . You don't learn much in a tax return. ''** Trump did release an extensive (and legally required) document detailing his personal financial holdings. However, experts consider that a red herring. Unlike all presidential nominees since 1980, Trump has not released his tax returns, which experts say would offer valuable details on his effective tax rate, the types of taxes he paid, and how much he gave to charity, as well as a more detailed picture of his income-producing assets. Trump's statement is inaccurate. **We rate it False.**

Table 13: Examples from L_{++} : the claim is mostly present within the first sentence, and a truthfulness statement is reported at the end of the verdict.

B Fine-tuning Details

For the fine-tuning, each model underwent 5 epochs of training with a batch size equal to 4 and a seed set at 2022. To this end, the Huggingface Trainer has been employed, keeping its default hyperparameter settings, with the exception of the Learning Rate values and the optimization method. The Adafactor stochastic optimization method (Shazeer and Stern, 2018) has been used throughout the whole training phase. Learning Rates values were set as follows: T5 3e-5, Peg$_{xsum}$ 5e-05, Peg$_{cnn}$ 3e-05, dBart 1e-05. For fine-tuning the models, we employed a single GPU, either a Tesla V100 or a Quadro RTX A5000. The checkpoint with minimum *evaluation loss* was used for testing.

C Extractive Approach Results Details

The first set of experiments tested three main text reduction methodologies: text truncation, LexRank, and SBERT. In order to assess the informativeness of the summaries, the generated extractive output was compared to the gold verdicts through ROUGE metrics. For each methodology, two main configurations have been taken into account: top and bottom (or head and tail for text truncation). While in Table 5 we just reported the best configuration (head/top), in Table 14 we report the complete results for extractive summarization, which includes the bottom configuration for comparison purposes. These results are confirmed also when these approaches are used for text reduction before the abstractive step in our pipeline.

	Extraction Method		R1	R2	RL
FF	truncation	head	0.258	0.082	0.182
		tail	0.216	0.047	0.149
	LexRank	top	0.267	0.083	0.180
		bottom	0.219	0.050	0.153
	SBERT	top	0.300	0.114	0.213
		bottom	0.178	0.030	0.132
L++	truncation	head	0.347	0.120	0.196
		tail	0.313	0.061	0.157
	LexRank	top	0.373	0.120	0.194
		bottom	0.302	0.056	0.154
	SBERT	top	0.393	0.158	0.237
		bottom	0.245	0.029	0.131

Table 14: Pure extractive approach results for the head/tail and top/bottom configurations. The number of sentences to be extracted is set to the average number of sentences per verdict in the corresponding datasets (2 for FF and 6 for L++).