Harnessing Abstractive Summarization for Fact-Checked Claim Detection

Varad Bhatnagar[1], Diptesh Kanojia[2,3], Kameswari Chebrolu[1]
[1]Department of Computer Science and Engineering, IIT Bombay, India
[2]Surrey Institute for People-Centred AI, [3]Department of Computer Science
[2,3]University of Surrey, United Kingdom
[1]{varadhbhatnagar,chebrolu}@cse.iitb.ac.in
[2,3]d.kanojia@surrey.ac.uk

Abstract

Social media platforms have become new battlegrounds for anti-social elements, with misinformation being the weapon of choice. Fact-checking organizations try to debunk as many claims as possible while staying true to their journalistic processes but cannot cope with its rapid dissemination. We believe that the solution lies in partial automation of the fact-checking life cycle, saving human time for tasks which require high cognition. We propose a new workflow for efficiently detecting previously fact-checked claims that uses abstractive summarization to generate crisp queries. These queries can then be executed on a general-purpose retrieval system associated with a collection of previously fact-checked claims. We curate an abstractive text summarization dataset comprising noisy claims from Twitter and their gold summaries. It is shown that retrieval performance improves 2x by using popular out-of-the-box summarization models and 3x by fine-tuning them on the accompanying dataset compared to verbatim querying. Our approach achieves Recall@*5* and MRR of 35% and 0.3, compared to baseline values of 10% and 0.1, respectively. Our dataset, code, and models are available publicly here.

1 Introduction

Social media is increasingly used for business, entertainment, and political discourse, thus, encouraging users to produce and consume large volumes of information that may not always be accurate. Due to a lack of digital awareness, the masses often believe and forward such disputed claims in their social circles. Such spread of misinformation often culminates in incidents which cause damage to life and property. It is well documented that misinformation is used as a tool by political agents to slander their opposition (Allcott and Gentzkow, 2017) and influence the opinion of the masses. It becomes furthermore dangerous when such claims pertain to religious beliefs, often leading to violence and mob lynchings[1]. In the era of COVID-19, unverified medical advice has also been circulated on social media (Shahi et al., 2021) which has already led to various health hazards.

Social media platforms have undertaken concerted efforts to tackle the fake news epidemic by enforcing strict policies to weed out unverified and sensitive content and ban habitual offenders. Journalistic organizations such as Alt News[2], Factly[3], Boom Live[4] and Snopes[5] among others are also fighting this problem by publishing fact-checking articles investigating the veracity of viral dubious claims. These articles detail the journalistic procedures followed to fact-check the claim along with suitable references.

Numerous researchers are working on AI-based solutions for fact-checking claims. Many datasets (Thorne et al., 2018; Sathe et al., 2020; Fan et al., 2020; Schuster et al., 2021) have been released to train models which can automate subtasks such as claim verification, evidence retrieval and assigning a verdict in a fact-checking workflow. A critical and insufficiently researched step in the fact-checking workflow is- *detecting whether a claim has been fact-checked previously*. This is a repetitive task with immense scope for automation, shrinking the turnaround time for a claim and ensuring that human efforts are put to better use on tasks involving higher cognition, such as assigning a verdict. In literature, learning-to-rank models (Shaar et al., 2020; Vo and Lee, 2020; Mansour et al., 2022) have been proposed for this step, which on being queried, produce a ranked list of results from a closed dataset of previously verified claims. Dozens of fact-checking articles are being published every hour around the world. It is

[1]Article on Mob Lynching: Washington Post
[2]AltNews: Website
[3]Factly: Website
[4]Boom Live: Website
[5]Snopes: Website

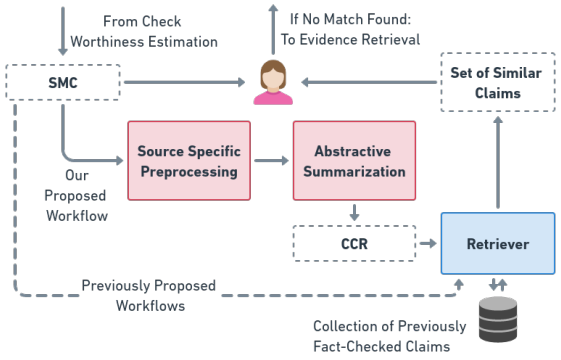


Figure 1: **Proposed Workflow**. In this work, we use Fact Check Explorer as a black box Retriever. The workflow proposed by previous works is denoted by a dotted path.

difficult for journalistic organizations to maintain such a large real-time collection of fact-checked articles and claims, thus, making such an approach infeasible in real-world scenarios.

We propose a novel workflow (as shown in Figure 1) for detecting previously fact-checked claims. In this work, we use Google's *Fact Check Explorer*, a cross-publisher, cross-language search engine for previously fact-checked articles, as a black box retriever. As social media platforms contribute a great deal to spreading misinformation, we deal with naturally occurring textual claims on Twitter in this study as opposed to artificial, well defined and structured claims (Thorne et al., 2018; Aly et al., 2021). Instead of querying using verbatim claims, which are noisy, it is proposed that abstractive text summarization be used as a precursor to querying to generate clear, succinct queries capturing the claim in a minimum number of words. Figure 1 represents a subpart of the complete fact-checking pipeline (Barrón-Cedeno et al., 2020) with our proposed changes in red. *CCR* and *SMC* are defined in Section 3. In literature, no dataset exists for abstractive summarization of tweets, and no attempts have been made to address this problem using the *Fact Check Explorer* to the best of our knowledge. Our contributions can be distilled into the following:

1. **Workflow**: A novel workflow for detecting previously-fact checked claims at scale.
2. **Dataset**: An abstractive summarization dataset[6] for tweets in the Indian context.
3. **Models**: Popular and large pre-trained abstractive summarization models, fine-tuned under supervision on this data, which can be used for other purposes involving tweet summarization.
4. **Experimental Study and Analysis**: We also perform quantitative and qualitative analysis for various outputs in our proposed workflow, including an analysis of generated summaries.

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 presents the dataset, Section 4 discusses the approach, evaluation metrics and the experimental setup. The results are presented and analysed in Section 5 followed by Section 6 which concludes the work and proposes future research directions. Section 8 and 9 discuss the ethical considerations and limitations.

2 Related Work

Sharma et al. (2019) describe the menace of misinformation on the Internet and summarize mitigation techniques and available datasets in this domain. Available intelligent technologies to assist the process of fact-checking are surveyed by Nakov et al. (2021a). This work highlights the partial overlap between current research endeavours and fact-checkers desiderata over the life cycle of a claim in a fact-checking pipeline.

A general-purpose four-step automatic fact-checking pipeline is presented by Barrón-Cedeno et al. (2020). The task of determining if a claim has been previously fact-checked is the second step in the pipeline. This problem is addressed in a series of open challenges (Shaar et al., 2021c) at *Checkthat!* workshop (Barrón-Cedeno et al., 2020; Nakov et al., 2021b, 2022) as part of CLEF [7]. Shaar et al. (2020) collect, annotate and release datasets of claim pairs and evidence sets, sourced from Politifact[8] and Snopes for solving this task. They develop and demonstrate the robustness of BM25 and BERT (Devlin et al., 2019) based learning to rank models on their dataset for this task. Vo and Lee (2020); Mansour et al. (2022) also propose variants of a ranking approach to solve this problem. Further, Shaar et al. (2021b) work with data from political debates and model the context of a claim and illustrate the positive impact this has in determining if it has been previously fact-checked. Shaar et al. (2021a) publish a dataset and develop a system for detecting all previously fact-checked claims in a lengthy document.

[6]Data and models are made available here: https://github.com/varadhbhatnagar/FC-Claim-Det/

[7]CLEF: Website

[8]Politifact: Website

Text summarization has been used to enable verdict explainability in automatic fact-checking (Mishra et al., 2020; Stammbach and Ash, 2020) but it hasn't been used for denoising tweets, to the best of our knowledge.

Tchechmedjiev et al. (2019) publish the *ClaimsKG* Knowledge Graph, containing 28K fact-checked claims and their metadata such as sources, truth value and entities. Structured queries can be executed on this knowledge graph, enabling exploration and information discovery. However, it does not provide any mechanism to check if a claim has been previously fact-checked. *Fact Check Explorer*[9] is a tool developed by Google which provides browsing and searching capability for already fact-checked articles which have the ClaimReview Schema[10] embedded. There are performance limitations associated with this tool in the face of long and complex queries.

3 Dataset

The following terms are defined for lucid perusal of this work:

1. **Social Media Claim (SMC):** A social media post (tweet, in this work) containing a claim in need of fact-checking. It is analogous to the output of the first step (*check worthiness estimation*) in the automatic fact-checking pipeline presented by Barrón-Cedeno et al. (2020).
2. **Fact Checked Article (FCA):** An article published by a fact-checking organization accepting or refuting a claim[11].
3. **Summary of Claim Review (SCR):** A short summary of the claim added by the publishing organization as part of the ClaimReview Schema associated with every FCA. Our use of the term SCR is the same as *VerClaim* coined by Shaar et al. (2020).
4. **Condensed Claim Representation (CCR):** A summary of the SMC generated using trained models.

3.1 Dataset Curation

In this work, we focus on FCAs published by Indian organizations between 2018 and 2022. FCAs from the following IFCN[12] certified organizations: 1) Alt News, 2) BoomLive, 3) India Today, 4) The Logical Indian, 5) The Quint, 6) Factchecker, 7) FactCrescendo, 8) Vishwas News, 9) PolitiFact, 10) Snopes, and 11) Factcheck.org, were retrieved. In order to make our dataset diverse, some FCAs from the USA based fact-checkers are also included, which shows that this workflow can be generalized.

Twint[13] is used to crawl Twitter, looking for URLs of the organizations mentioned above, in the comment threads of tweets. This resulted in a coarse collection of *<Tweet, SCR>* pairs. Those pairs with tweets in languages other than English and tweets containing only image/video content are discarded. We perform annotation on this collection, keeping two aspects in mind: (1) the tweet should contain a claim, and (2) it should be textually summarizable to the corresponding SCR. URL removal from SMCs followed by pairwise de-duplication is performed at this stage, resulting in our final dataset, a collection of *<SMC, SCR>* pairs, which can be used for training abstractive text summarization models. *The final dataset only contains <SMC,SCR> pairs where both are in English.*

Key world and Indian events have been covered as part of this dataset, such as the onset of COVID-19 and subsequent immunisation, the Taliban takeover of Afghanistan, Indian General Elections 2019 and US Presidential Elections 2020. Our annotation process is detailed below.

3.1.1 Annotation Details

Two trained annotators were tasked with annotating every *<Tweet,SCR>* pair from the coarse collection (Subsection 3.1). Three categorical attributes viz. Tweet language, SCR language, category and one boolean attribute *viz.* 'Summarizability' had to be populated for each pair.

The annotators were provided instructions to mark a pair as 'summarizable' only when the SCR is a condensed version of the tweet and named entity coverage is more than 50%. For deciding entity coverage, the annotators were allowed to take cues from the mentions and hashtags in the tweet. As majority of the FCA Publishers we dealt with are Indian, a lot of tweets and SCRs were in Indian languages such as Hindi, Hindi transliterated in English, Tamil, Telugu, and some other Indian languages. Any such instances were pruned from our dataset.

[9] Fact Check Explorer: Web Search
[10] ClaimReview Schema
[11] Example FCA
[12] IFCN: Website
[13] Twint: Github Repository

To understand the motivation behind these SMCs, our annotators were also requested to categorize them into classes like a) Politics, b) Crime and Terrorism, c) World, d) Entertainment, e) Technology, f) Food, g) Religion, h) Sports, i) Health, j) Education, k) Business, l) Environment, and m) Other (miscellaneous). Though not relevant to this work, nor a part of the final dataset, we collect and annotate this data as well for further research.

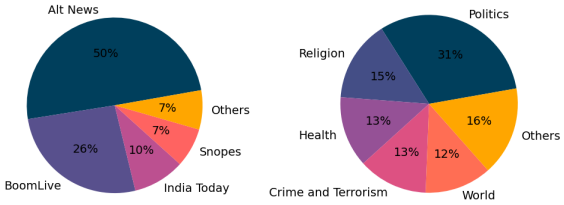
We observe an inter-annotator agreement of 92% within the annotations provided by both.

3.2 Dataset Statistics

Data Entity	**Count**		
<*SMC,SCR*> pairs	567		
Unique SMC	531		
Unique SCR	369		
FCA Source Country			
India	93%		
US	7%		
Median Length	**Chars**	**Words**	
SMC	193	33	
SCR	70	11	
Data Sets	**Cosine Similarity Threshold**		
	0.25	**0.5**	**0.75**
NP	59%	12%	2%
P-H-M	61%	13%	3%
Snopes (Shaar et al., 2020)	50%	8%	1%

Table 1: **Dataset Statistics and Complexity Analysis**. NP and P-H-M are defined in Subsection 4.1.

The statistics of our final dataset, comprising of 567 unique <*SMC, SCR*> pairs are presented in Table 1. Owning to several tweets and several FCAs about the same underlying event, 1:1 correspondence is not observed in the dataset, as evident from the first section in this table. Due to the 280 character limit imposed on tweets by Twitter, the SMCs are not arbitrarily long, with a median length of 33 words and the SCRs are observed to be very short, with a median length of 11 words. Similar to (Shaar et al., 2020), the complexity of the task is analyzed by reporting the word-level TF-IDF weighted cosine similarity for <*SMC, SCR*> pairs. Since our dataset supports summarization, cosine similarity is higher compared to the Snopes dataset by (Shaar et al., 2020), as expected. Figure 2 presents the FCA Source and SMC Topic distribution. 46% of the SMCs are political or religious, which is no surprise as these sensitive topics polarise opinion very easily. A large chunk of SMCs are health-related, owning to misinformation surrounding the COVID-19 immunization and mass hysteria.



(a) FCA Sources (b) SMC Topics

Figure 2: **Dataset Distribution**

4 Our Approach

SMCs are very noisy in nature due to the inherent way people interact on social media and micro-blogging platforms. On Twitter, tweets are bounded by a character limit, forcing people to use slang and abbreviations to communicate effectively. It also allows for mentions and hashtags to be embedded in tweets to encourage inter-user interaction. Using these noisy SMCs verbatim (as done by Shaar et al. (2020)) to check if they have been previously fact-checked is challenging, as the retrieval module has to do all the heavy lifting for which it is not equipped.

In this work, it is hypothesized that a system which extracts queryable content from SMCs by dealing with its syntactic and semantic aspects before querying the retrieval module should perform better than verbatim querying. Keeping in mind the small scale at which fact-checking organizations work and the continuously growing collection of FCAs, Google's *Fact Check Explorer* is used as a retriever for previously fact-checked claims instead of a closed collection of verified claims. The *Fact Check Explorer* indexes the latest FCAs across the world and provides easy to use search APIs for free, which support filtering based on publisher and language, among other features. Various text preprocessing techniques on SMCs are experimented with before using state-of-the-art abstractive text summarization models to generate corresponding CCRs. These CCRs are then used to query the retriever. These techniques and the models used are detailed in the following subsections. Our prefer-

ence for abstractive summarization over extractive summarization arises because of two reasons; the SMCs are noisy and unlikely to contain query-able spans and due to the recent progress in abstractive summarization research (Lewis et al., 2019; Zhang et al., 2020; Raffel et al., 2020; Xiao et al., 2021).

This proposed workflow is generic in nature as <*SMC, SCR*> pairs collected from other microblogging platforms (using our curation methodology) can be used to train summarization models after applying text pre-processing techniques specific to that platform. These models can also generate queries for open-domain evidence retrieval, which is the next step in a fact-checking pipeline. It is also futuristic in the sense that a generative module can replace the text summarization module to support multimodal SMCs. However, this work is kept limited to textual SMCs due to a lack of suitable labelled data and the absence of a reliable equivalent of *Fact Check Explorer* for joint text, image and video search.

4.1 Twitter Specific Preprocessing

Most social media platforms encourage inter-user interaction by allowing 'mentioning' other users in a post. Typically, some form of notification goes to the user being mentioned, getting his attention on the post content. It is also used as a way for tagging people to establish their presence in photos and videos. Hashtags are metadata tags which allow cross referencing of content by topic or theme. They typically identify with some event or social movement, allowing users to discover and associate with trending content. Both hashtags and mentions are available on Twitter along with emojis, which are smileys embedded in text, providing emotional cues to the reader.

We experiment with these three aspects of a tweet. Most search engines do not deal with Emojis, hence we replace them with a constant to form the P+MRep set. Hashtags and Mentions provide rich signals about named entities, hence it is important to incorporate them in the input in some way. Upon manual analysis of the data, it was seen that a lot of tweets mentioned users who were unrelated to the content in the tweet. Some recurring instances of this phenomena that we came across, were fact-checking requests mentioning many journalists and organizations and political tweets mentioning prominent members of the opposition political party and prominent believers of the opposite ideology. It was observed that hashtags were also used in a similar manner. Another observation was the existence of runs of space separated hashtags and mentions and their occurrence at the beginning or end of the tweet, signifying the preference of users to separate actual tweet content from these meta tags. These signals led us to create sets of data where mention and hashtag runs are removed except the first member in each run (P-MRR-HRR). Further, some users used organization related twitter handles or twitter handles in other languages like Hindi. To deal with this, we replace these by their original names on Twitter to get the P-MRR-HRR+MRep set. With a clear intuition behind such preprocessing, we now describe what Twitter specific text preprocessing techniques are applied to SMCs to produce the following <*SMC,SCR*> sets, from the final dataset:

1. **Verbatim (NP)**: SMCs are used verbatim.
2. **Preprocessed (P)**: Symbols for hashtags(#) and mentions(@), emojis, punctuation and redundant are removed, followed by lowercasing of SMCs.
3. **Pre-processed with Emojis Replaced (P+ERep)**: Emojis are replaced by the string *$EMOJI$* in addition to techniques used in P.
4. **Pre-processed with Hashtags and Mentions Removed (P-H-M)**: All hashtags and mentions are removed in addition to techniques used in P. Subsets with only hashtag removal **(P-H)** and only mention removal **(P-M)** are also created.
5. **Pre-processed with Mention and Hashtag Run Removed (P-MRR-HRR)**: Run of hashtags and mentions are removed, except the first entity in each run, in addition to techniques used in P.
6. **Pre-processed with Mention and Hashtag Run Removed and Mentions Replaced (P-MRR-HRR+MRep)**: The remaining mentioned handles in P-MRR-HRR are replaced by their official names from Twitter.

4.2 Summarization Models

For summarization, the following models were experimented with:

1. **Truncate *k***: A naive summarizer which truncates a SMC to the first k space-separated tokens. It is used as a baseline to show gains by more complex models.

2. **T5**: A transformer-based architecture by Raffel et al. (2020) that uses a text to text approach for all tasks. It is pre-trained on a multi-task mixture of supervised and unsupervised tasks such as denoising on the high quality C4 corpus, sentiment analysis, natural language inference and question answering, among others. To make the model cope with this multi-task training, a task-specific prefix is added to the input sentence.
3. **BART**: A transformer-based sequence to sequence model by Lewis et al. (2019) which incorporates the bidirectional encoder of BERT (Devlin et al., 2019) and the left-to-right autoregressive decoder of GPT (Radford and Narasimhan, 2018; Radford and Wu, 2019), pre-trained in denoising autoencoder style. It works well for downstream tasks involving text generation.
4. **PEGASUS**: A transformer-based sequence to sequence model by Zhang et al. (2020) which uses a self-supervised pre-training objective called gap-sentence generation, aimed at optimizing downstream abstractive summarization tasks. In gap-sentence generation, important sentences in a document are masked, and the transformer model is asked to predict those sentences. PEGASUS shows impressive performance even with a small number of samples during fine-tuning.

4.3 Decoding Strategies

Decoding strategies define how text should be generated by models that support language generation. Based on the end application, the model may be expected to generate text that can be lengthy, short, non-repetitive, interesting, surprising, and so on. Our application requires the output to be short and crisp. In this work, we experiment with Greedy, Beam Search, Top k and Top p (Holtzman et al., 2019) decoding strategies.

4.4 Evaluation Metrics

Recall@k and Mean Reciprocal Rank (MRR) are reported for all experiments, as is the norm in retrieval tasks. While checking if a claim was previously fact-checked, fact-checkers would not want to look beyond the first few results. Keeping this in mind, Recall@*5* is used as the primary metric for comparing retrieval performance. Figure 3 shows the variation in Recall@k with increasing value of k. The sharp bend at Recall@*5*, subsequent plateauing also motivated us to report this metric for retrieval. Also, it is practically feasible for a human fact-checker to go through 5 results per claim rather than 10 or 20 results.

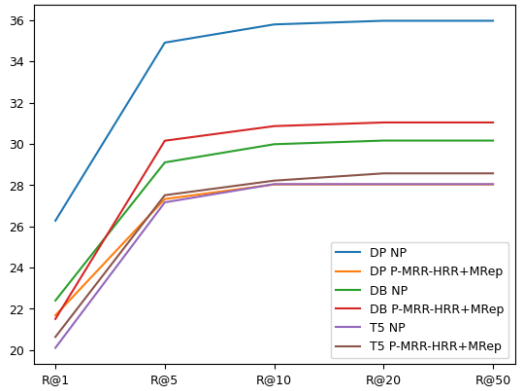


Figure 3: **Recall Plateauing for Decoding Strategies**

For evaluating the quality of the summary generated, word-level TF-IDF weighted cosine similarity between SMCs and CCRs and between SCRs and CCRs is reported. BLEU4 (Papineni et al., 2002).

4.5 Experiment Setup

In the experiments, the performance of summarization models in both out-of-the-box settings and through fine-tuning, *i.e.,* training on the task under supervision are compared. For the fine-tuning experiments, 5-fold cross-validation is performed on the data; and mean values along with the standard deviation are observed. Other experiments are performed on the entire data without any splits as no parameter learning is involved.

All experiments performed with the help of Transformer-based architectures in Table 2 use Beam Search decoder with a beam size of 6 and the maximum token length of a generated sequence, set to 15 with early-stopping enabled. For Truncate k experiments, k=11 is set. We arrive at these constants by looking at the median summary length provided in Table 1 and giving some leeway to transformer models as they operate on sub-word vocabularies. Hugging Face[14] implementations of the models mentioned in Subsection 4.2 are used for all experiments involving transformers. For the PEGASUS and BART experiments, we use the distilled versions released by Shleifer and Rush (2020) using the shrink and fine-tune approach on the CNN dataset. The 16-layer-encoder 4-layer-

[14] Hugging Face: Website

	No Summarization				Summarization using Out of the Box Models						Summarization using Fine Tuned Models					
Preprocessing Strategies	**None**		**Truncate11**		**T5**		**D BART**		**D PEGASUS**		**T5**		**D BART**		**D PEGASUS**	
	R@5	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**	**R@5**	**MRR**
NP	9.52	.09	17.28	.14	15.52	.13	20.46	.17	**22.40**	**.19**	27.16 ±2.55	.23 ±.02	29.10 ±3.15	.26 ±.02	**34.91 ±5.91**	**.30 ±.05**
P	11.99	.11	18.34	.15	17.64	.15	17.99	.14	21.52	.17	28.38 ±6.55	.24 ±.05	28.21 ±6.88	.24 ±.06	27.69 ±2.63	.24 ±.02
-H	12.70	.12	17.99	.15	17.28	.15	17.64	.14	20.46	.16	24.87 ±5.13	.21 ±.04	30.15 ±6.08	.26 ±.06	29.61 ±6.59	.25 ±.05
-M	13.05	.12	18.69	.15	18.34	.15	17.64	.14	21.16	.17	26.80 ±5.33	.23 ±.03	*30.15 ±4.98*	.26 ±.04	29.10 ±2.45	.25 ±.02
-H-M	13.93	.13	**18.87**	**.15**	17.81	.15	17.28	.14	20.28	.16	26.46 ±4.69	.23 ±.03	27.51 ±4.09	.23 ±.04	26.28 ±2.36	.23 ±.02
+ERep	10.58	.10	17.46	.14	16.58	.14	17.99	.15	22.05	.18	27.51 ±6.13	.23 ±.05	28.20 ±5.89	.25 ±.05	30.15 ±5.72	*.27 ±.05*
-MRR-HRR	12.35	.11	17.81	.15	17.46	.15	17.81	.14	21.69	.18	28.75 ±6.38	.25 ±.05	27.33 ±5.94	.23 ±.05	25.92 ±1.98	.22 ±.01
-MRR-HRR +MRep	12.70	.12	**18.87**	**.15**	18.34	.15	17.81	.14	21.87	.18	27.51 ±5.43	.24 ±.04	30.15 ±5.98	.25 ±.05	27.32 ±3.87	.24 ±.03
Skyline	**63.85**	**.55**														

Table 2: **Retrieval Results** (Subsection 5.1). D BART and D PEGASUS stand for Distilled BART and Distilled PEGASUS respectively, and Recall@*5* is represented by R@5. All preprocessing strategies prefixed with '+' or '-' are applied on top of the P set.

decoder version of distilled PEGASUS[15] and 12-layer-encoder 6-layer-decoder version of distilled BART[16] are used. The base version of T5[17] is used for all experiments involving T5. The number of trainable parameters are 220M, 300M and 370M in T5-base, Distilled BART and Distilled PEGASUS, respectively. Since *Fact Check Explorer* is an ever-growing and evolving system, CCRs are generated for all experiments first, and then retrieval queries are run, ensuring consistency across results. For retrieval, the API documentation[18] is followed and retrieved URLs are compared with normalized (removing redirection/parameters accompanying the URL) URLs associated with an FCA.

5 Results and Discussion

This section discusses the results obtained at various stages of our workflow.

5.1 Retrieval

We present the retrieval results in Table 2. From top to bottom, the SMC pre-processing (Section 3) complexity increases and from left to right, the complexity of the summarization models (Subsection 4.2) increases. For the skyline numbers, *Fact Check Explorer* is queried using the gold SCRs, giving 63.85 Recall@*5* and 0.55 MRR. We observe two evident trends via experimentation- (1) the performance gain in using an out-of-the-box summarization model, as compared to no summarization, and (2) the benefit of learning under supervision on our labelled dataset, indicated by the sharp gain in performance of fine-tuned models as compared to the corresponding out of the box models. Since the PEGASUS model is pre-trained with an objective to boost abstractive summarization performance, it works quite well out-of-the-box, giving a 2x increase in performance compared to no summarization. The best performing model is Distilled PEGASUS, fine-tuned on our dataset (without any pre-processing), as exhibited by a Recall@*5* of 34.91 and MRR of 0.3, which is more than 3x improvement over verbatim querying.

We use three different summarization strategies- (1) No Summarization, (2) Summarization using out-of-the-box Models, and (3) Summarization using fine-tuned models as shown in Table 2. We separately highlight the best performance in the table itself in each of these cases. In the no summarization experiments, we observe that complex pre-processing techniques lead to a performance gain, as indicated by the best Recall@*5* and MRR of 18.87 and 0.15 on dealing with mentions and hashtags (for both P-H-M and P-MRR-HRR+MRep). Among the out-of-the-box experiments, it is seen that Distilled PEGASUS comfortably outperforms T5 and Distilled BART, with the best Recall@*5* and MRR being 22.4 and 0.19, respectively. Highly parameterized models like BART and PEGASUS do not benefit from input pre-processing.

The gap between the skyline numbers and the best performing model can be attributed to the fact that most models are pre-trained on document level summarization datasets such as CNN/Daily Mail (Nallapati et al., 2016) and Huge News (Zhang et al., 2020). Hence, they struggle with summarizing short input text.

5.2 Summarization Quality

The quality of CCRs is reported in Table 3. We compare CCRs with SMCs and SCRs on two metrics (1) Word level TF-IDF Weighted Cosine Simi-

[15] Distilled PEGASUS Model
[16] Distilled BART Model
[17] T5-base Model
[18] Fact Check Explorer: API Documentation

Experiment	n-Gram	Cosine Similarity Threshold			BLEU4
		0.25	0.5	0.75	
E1 SMC vs CCR	1	80%	26%	2%	-
	2	83%	30%	2%	-
E1 SCR vs CCR	1	76%	38%	14%	39.7
	2	73%	38%	13%	39.3
E2 SMC vs CCR	1	83%	26%	4%	-
	2	85%	29%	3%	-
E2 SCR vs CCR	1	68%	31%	11%	38.9
	2	69%	33%	13%	39.2

Table 3: **Summarization Quality Analysis** (Subsection 5.2). E1 and E2 stand for Distilled PEGASUS with NP and P-MRR-HRR+MRep experiments respectively (as described in Section 4) and n-Gram stands for the value of n in n-grams not appearing more than once in beam search decoding.

larity and (2) BLEU4. Since BLEU4 is generally reported between reference and generated sequences, it does not make sense to report it for SMC vs CCR rows. *We observe high BLEU4 scores for the CCRs* signifying that our approach can generate valid summaries, as can also be seen in Table 4. For both E1 and E2, our BLEU4 scores are approaching (approx.) 40. On comparing SMC vs CCR cosine similarities for both experiments with the cosine similarity of SMC vs SCR (last section in Table 1), we find higher values for all thresholds indicating that *our generated summaries are significantly similar to the tweets as compared to the gold summaries provided.*

5.3 Decoding

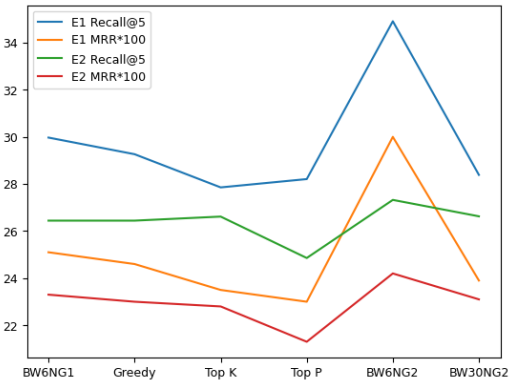


Figure 4: **Decoding Strategy Comparison** (Subsection 5.3). MRR values are multiplied by 100 for better visualization. BWkNGn corresponds to beam search decoding with beam width k and no n-grams appearing more than once in the generated output. k is set to 50 in Top k and p is set to 0.92 in Top p.

Figure 4 shows the variation in retrieval results on using different decoding strategies. Definitions for E1 and E2 follow from Table 3.

As observed from this figure, BW*6*NG*2* seems to be the best performing decoding strategy. Hence, this strategy is used for all experiments in Table 2. BW*6*NG*1* also seems to be a good alternative, but the *1*-gram constraint makes the queries very terse and grammatically inconsistent (observed manually). Greedy, Top k and Top p strategies are not competitive for such a task.

5.4 Larger Language Models

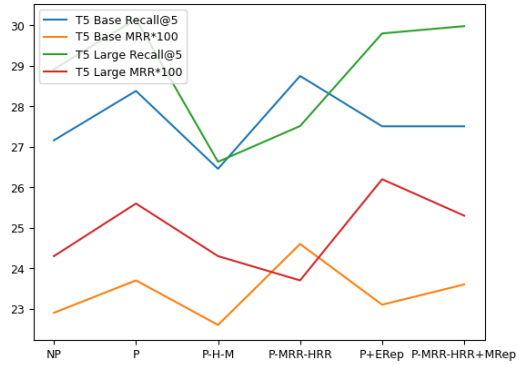


Figure 5: **Effect of Larger Language Models on Retrieval Metrics**

We study the variation in performance using even larger models such as T5 Large[19], which has 770M parameters, three times that of T5-base. CCRs generated by the larger model perform better on both retrieval metrics across a variety of preprocessed SMCs, but the performance gain is not significant. It is offset by a longer training time and heavy compute requirements leading to considerable cost overheads. Since this is not a study of large generative models and given the modest resources owned by most fact-checking organizations, we do not explore any larger language models such as T5-3B and T5-11B, which have 3 Billion and 11 Billion parameters, respectively.

5.5 CCR Quality

Table 4 lists a few CCRs generated by the best performing model, also listing the corresponding SMCs and SCRs. The model successfully extracts the core claim from the SMCs and ignores tokens like mentions and hashtags that have no contribution to the core claim. Owning to the constraints placed on length, it is seen that the generated CCRs are succinct and context-independent. They seem to be paraphrases of the gold SCRs, making them

[19] T5-large Model

#	SMC	SCR	CCR
1	Congratulations to Uttarakhand CM for becoming the first CM ever to charge stranded citizens for rescue operations! Helicopter rides will now be chargeable during rescue operations in Uttarakhand. And if you can't pay, you may safely die. #AchheDin #BJP	Passengers in Uttarkhand to be charged for rescue operations	Uttarakhand CM has charged stranded citizens for helicopter rides during rescue operations
2	@AltNews We are getting various WhatsApp forward regarding as Corona has been emerges only due to 5G testing in world. Please put some light, seems ,it is only a brain shit.	5G radiation is the cause behind the second wave of coronavirus pandemic in India	Coronavirus outbreak due to 5G testing
3	This woman in Afghanistan was killed by Taliban for not wearing the proper cloth. #Afghanistan #Taliban @cnn @FoxNews @BBCWorld	Video shows a woman being shot in the head by Taliban in Afghanistan for not dressing appropriately	Woman killed by Taliban for not wearing proper cloth
4	"India is ranked 102nd in the global hunger index, out of 117 countries. We are ranked in between Niger & Sierra Leone. We are the lowest ranked South Asian country. Bangladesh is ranked 88th and Pakistan 94th. They have only recently overtaken us. Our rank was 55,only 5 years ago"	India's ranking in Global Hunger Index (GHI) has fallen from 55 in 2014 to 102 in 2019	India ranked 102nd in the global hunger index
5	"Oxygen donated from Saudi and relabelled in india by Reliance, Share this with your contacts in Saudi and make this viral .. Let the world know the cheapness of this PM "	Oxygen sent from Saudi Arabia is being distributed in the name of Reliance	Reliance taking credit for oxygen supplied by Saudi Arabia

Table 4: **SMCs and SCRs from the Dataset with corresponding CCRs** (Subsection 5.5).

good candidates for querying the retrieval system. It is also seen that our model finds factual inputs which require reasoning, difficult to deal with. For instance, #4 in Table 4 requires a model to understand that going from rank 55 to 102 in the Global Hunger Index is a fall and not a rise. Our workflow does not expect the underlying language model to understand and reason, and this workflow only requires the generation of a valid summary.

6 Conclusion and Future Work

In this work, a new workflow for detecting previously fact-checked claims is proposed. This workflow uses text summarization as an intermediate step before retrieval module invocation. Clean and crisp summaries thus generated are then used for querying a retrieval system. To this end, a first-of-its-kind tweet summarization dataset in the Indian context to train such models is curated and released under the CC-BY-NC-SA 4.0 license. The performance gained on using popular out-of-the-box and fine-tuned summarization models before querying the *Fact Check Explorer* is demonstrated, and discussed with qualitative samples. Various popular decoding strategies are compared, and the implication of using larger pre-trained models is explored. *The aim of this work is to aid in the creation of general-purpose and performant modules which can speed up a fact-checking pipeline by equipping fact-checkers with the tools to fight misinformation at a large scale.*

In future, we would also like to perform this task in a more general context for news items from various countries, extending our work in a multilingual scenario. Also, named entities are crucial in drafting a good query for any retrieval system. Generating summaries based on the Named Entities (Zhang et al., 2020) found in SMCs is a promising avenue to explore. We do not take tweet threads into account as our focus is SMCs by users and not replies or comments to those SMCs, however, this can be an interesting future direction. Other controlled text generation (Keskar et al., 2019; Chan et al., 2021) techniques can also be explored to extract the maximum information from noisy SMCs. Better pre-training objectives for abstractive summarization on noisy text can lead to efficient out-of-the-box models for this task.

Most Indian fact-checking organizations in Section 3.1.1 also publish FCAs in regional languages such as Hindi, Tamil and Telugu. Twitter conversations, spreading misinformation in other pure and transliterated Indic languages are voluminous. Cross-lingual summarization research (Zhu et al., 2019) would go a long way in fighting misinformation in a holistic manner.

7 Acknowledgements

We acknowledge the kind support provided by IMPRINT-2, a technology development initiative of MHRD and DST, Government of India.

8 Ethical Considerations

To the best of our knowledge, no code of ethics was violated throughout the experiments performed for this study. We report all hyper-parameters and other technical details necessary to reproduce our results, and release the code and dataset curated via this work. We perform our experiments with the help of various language models which may contain biases as discussed by Weidinger et al. (2021). However, we believe that our workflow and methodology are solid and apply to any social media fake news setting. Any quantitative results reported by us are reproducible, subject to the ever growing number of articles indexed by the Fact Check Explorer (reported in Section 4.5). However, the qualitative results (like generated summaries) are an outcome of computational models that does not represent our personal views. We do not include any identifying information in the data that we use for our experiments and ensure that the dataset release will follow anonymization of any such information.

We would like to state that this dataset is collected in a recent real-world setting (raw social media claims from 2018-2022) and *no attempt has been made by us to subdue tweets on certain topics and promote others*. More precisely, we freely assigned the tweets to our annotators without any domain/topic specificity, however, they were required to label the tweet from a list of categories (Section 3.1.1) to collect more information.

9 Limitations

We believe there is a limitation to our work, *i.e.,* **The limited size of this dataset**; which can be attributed to following reasons:

- Most fact-checking organisations (covered in this work) emerged post-2017.
- Our data curation relies on a large number of users replying to potentially misinformative tweets. This user behaviour is limited by social network usage, awareness and internet proliferation for a particular language, region or country.
- The "manual pruning" step while curating the tweet level summarization dataset was a very time/effort-intensive process. For e.g., around 5000 coarse <*Tweet,SCR*> pairs were manually pruned to get the final dataset containing 567 <*SMC,SCR*> pairs, implying a rejection rate close to 90%.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. Working Paper 23089, National Bureau of Economic Research.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Alberto Barrón-Cedeno, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, and Fatima Haouari. 2020. Checkthat! at clef 2020: Enabling the automatic identification and verification of claims in social media. In *European Conference on Information Retrieval*, pages 499–507. Springer.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2021. Cocon: A self-supervised approach for controlled text generation. In *International Conference on Learning Representations*.

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.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7147–7161, Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

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*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Watheq Mansour, Tamer Elsayed, and Abdulaziz Al-Ali. 2022. Did i see it before? detecting previously-checked claims over twitter. In *European Conference on Information Retrieval*, pages 367–381. Springer.

Rahul Mishra, Dhruv Gupta, and Markus Leippold. 2020. Generating fact checking summaries for web claims. In *EMNLP W-NUT 2020: Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, et al. 2022. The clef-2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection. In *European Conference on Information Retrieval*, pages 416–428. Springer.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barr'on-Cedeno, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021a. Automated fact-checking for assisting human fact-checkers. In *IJCAI*.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeno, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, et al. 2021b. The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *European Conference on Information Retrieval*, pages 639–649. Springer.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gu̇lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. In *Published by OpenAI*.

Alec Radford and Jeffrey Wu. 2019. Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI Blog*, 1(8):9.

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.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from Wikipedia. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France. European Language Resources Association.

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.

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2021a. Assisting the human fact-checkers: Detecting all previously fact-checked claims in a document. *arXiv preprint arXiv:2109.07410*.

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2021b. The role of context in detecting previously fact-checked claims. *arXiv preprint arXiv:2104.07423*.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Shaden Shaar, Fatima Haouari, Watheq Mansour, Maram Hasanain, Nikolay Babulkov, Firoj Alam, Giovanni Da San Martino, Tamer Elsayed, and Preslav Nakov. 2021c. Overview of the CLEF-2021 CheckThat! lab task 2 on detecting previously fact-checked claims in tweets and political debates. In *Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum*, CLEF '2021, Bucharest, Romania (online).

Gautam Kishore Shahi, Anne Dirkson, and Tim A. Majchrzak. 2021. An exploratory study of covid-19 misinformation on twitter. *Online Social Networks and Media*, 22:100104.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):1–42.

Sam Shleifer and Alexander M Rush. 2020. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*.

Dominik Stammbach and Elliott Ash. 2020. e-fever: Explanations and summaries for automated fact checking. *Proceedings of the 2020 Truth and Trust Online (TTO 2020)*, pages 32–43.

Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. Claimskg: a knowledge graph of fact-checked claims. In *International Semantic Web Conference*, pages 309–324. Springer.

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.

Nguyen Vo and Kyumin Lee. 2020. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *CoRR*, abs/2112.04359.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. 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 3054–3064, Hong Kong, China. Association for Computational Linguistics.