UHH Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Master Thesis

Fake News Detection with Journalists-in-the-Loop

Soniya Vijayakumar

MIN-Faculty
Language Technology Group

Degree program	:	M.Sc. Intelligent Adaptive Systems
Matriculation number	:	7133603
E-Mail	:	8vijayak@informatik.uni-hamburg.de
Supervisors	:	Professor Dr. Chris Biemann
	:	Dr. Seid Muhie Yimam

Submission Date: 24.06.2021

Abstract

Computational detection of fake news has gained popularity since the spread of fabricated and false information on the online media platform. On one side, there exist many human-based fact-checking sites (e.g., snopes.com and politifact.com) that try to check for the veracity of claims. And on another side, many automated fact-checking systems are being developed using state-of-the-art deep learning models. Fact-checking websites are completely dependent on manual verification whereas, automated systems rely on the ability of natural language processing methods. There exists a gap between these two systems, which is the lack of inclusiveness of the end-user in the automated detection. We present a unified end-to-end fake news detection framework that comprises of computational detection of fake news with a journalist-in-the-loop approach. Furthermore, we observe that explainability in human-understandable form is crucial in such frameworks. With an objective to create an assistive framework for journalists, we generate justifications and present online sources as evidence. We also create a new knowledge base, fine-tune deep learning models using various datasets, and present qualitative and quantitative evaluation results of the framework. Quantitative evaluations indicate a new baseline for the veracity prediction model and qualitative feedback from user study confirms the utility of such a framework and its contemporary relevance.

Acknowledgement

I would like to take this opportunity to thank my supervisor Dr. Seid Muhie Yimam for the immense support he has extended to me in every step of this thesis. During these pandemic days, where meetings are only virtually possible, he has always been patient with me in clarifying all my doubts and guiding me to the right directions.

I would also like to express my sincere gratitude to Prof. Dr. Chris Biemann for giving me this opportunity to work on one of the most relevant topics in the Language Technology domain.

A special acknowledgement to the CheckFirst team[1] for supporting and guiding with relevant inputs and feedbacks from the journalists perspective.

Last but not the least, I would like to thank the Language Technology group for providing me with the infrastructure requirements for implementation as well as providing valuable feedbacks during the final presentation.

My parents and my closest friends have always extended unconditional support in all my academic ventures and I am grateful to them for their continuous encouragement and motivation.

[1] https://checkfirst.network/

Contents

Chapter 1

Introduction

Online media has opened up to a plethora of information from all spheres of social life, leading to transformational trends in news-making. News-making has moved from a few traditionally dominant news organizations to a multitude of social media users, hence leaving no control over the veracity of the information that is being published every day. With such an open social media ecosystem and self-publishing popularized among people, the question that lingers in one's mind is "How much of what we read on the internet is actually information or misinformation?"

Furthermore, this challenge becomes extremely acute during notable events like natural disasters, political turmoils, pandemics like the current COVID-19, due to the expedited reporting of new findings. Along with misinformation, another largely spread challenge is the existence of Fake News, not just in the online media but within the social norms of societies as well. The first basic understanding one requires in such context is the differentiation between misinformation and fake news.

The Ethical Journalism Network[1] defines fake news as: "Fake news is information deliberately fabricated and published with the intention to deceive and mislead others into believing falsehoods or doubting verifiable facts"

According to the Council of Europe's Information Disorder Report of November 2017[2], there are three types of information disorder. Much of the discourse on 'fake news' conflates three notions: misinformation, disinformation, and malinformation.

- **Disinformation:** Information that is false and deliberately created to harm a person, social group, organization, or country.

- **Misinformation:** Information that is false, but not created with the intention of causing harm.

- **Malinformation:** Information that is based on reality, used to inflict harm on a person, organization, or country.

[1] https://ethicaljournalismnetwork.org/tag/fake-news/page/5

[2] https://rm.coe.int/information-disorder-report-november-2017/1680764666

The difference between these notions is evidently based on their intentions; disinformation or hoaxes are popularly referred to as 'Fake News'. Going forward in this thesis, we will use the term 'claim' to refer to any information that requires to be verified for its truth.

1.1 Motivation

The spread of fake news is not a new concept. Before the digital era, the spread was confined to yellow journalism, which focused mainly on sensational news such as crime, gossip, disasters, and satirical news (Potthast et al. (2018)). With the new era of social media and its nature, it has become so easy to spread fake news, making fake news detection a complex challenge. Our daily lives rely on information and if this information is deliberately created to be fake, misleading, or exaggerated, they cause a huge amount of impact in various directions like health, finances, psychological, democratic, and political impacts.

Another multitude of consequences that have aroused from such open systems is the increased tediousness faced by journalists for verifying published information. Currently, their efforts are non-exhaustive in nature, involving manual exploration of online resources and existing databases. In this manual fact-checking framework, as evidently stated by Atanasova et al. (2020), justifications of claim verification are arguably also an important part of the journalistic processes.

Considering the harmful impacts of fake news spread, the complexity involved in detecting them, and with a unique objective to assist journalists in this verification and debunking process, this thesis proposes to develop a framework with the following objectives: 1) An automatic and adaptive fake news detection application that outperforms itself over time; 2) An effective fake news detection pipeline as a framework inclusive of journalists and, 3) An explanation-generating system, inclusive of manual-explanations, proposing the reasons for claim veracity.

1.2 Research Questions

With such an existing real-time challenge faced by journalists and with the common users facing a plague of falsehood proliferating around the internet, the research questions formulated are as follows:

- **Research Question 1 (RQ1):** What will be an effective architecture for end-to-end fake news detection pipeline as an assistive system for journalists?

- **Research Question 2 (RQ2):** How can such a system ease the fact-checking efforts of journalists: From a journalist fact-checking framework, what factors are important for easing the users' efforts in fact-checking?

- **Research Question 3 (RQ3):** What is a considerable accuracy of the deep learning model (54+%) for the entire pipeline to be considered effective for journalists?

- **Research Question 4 (RQ4):** How is the veracity of a claim arrived at: What information is to be extracted from the pipeline's component to arrive at a human-understandable explanation?

RQ 1 and 2 focuses on including the expertise and domain knowledge from journalists. RQ3 focuses on building an effective veracity prediction and veracity explanation framework components using natural language processing and deep learning models. RQ4 aims at providing reasons for the predictions made by the automated fake news detection application. With these research directions in mind, the following are the objectives of this thesis work:

1. Create a comprehensive knowledge base from existing as well as newly collected data using continuous news retrieval techniques.
2. Develop a fake news detection framework using NLP techniques and deep learning approaches.
3. Build a web-based interactive user interface to record feedback from journalists into the system.

1.3 Contributions

The major contributions we make by implementing the **Fake News Detection Framework** are as follows:

1. An effective open-source end-to-end framework that allows end-users to provide feedback.
2. A new knowledge base with multi-dimensional features.
3. Inclusion of veracity prediction and explanation within the same framework.

Finally, the datasets and the source code are publicly available to advance further research in fake news and misinformation intervention.

The remaining of this thesis is organized as follows: Chapter 2 describes the relevant literature review and Chapter 3 explains the theoretical aspects of the deep learning architectures used in implementing our framework. Chapter 4 and 5 is about creation of knowledge base and the respective datasets used for fine-tuning the models. This chapter also presents an analysis of these datasets to understand their features. Chapter 6 presents the framework itself in detail by explaining the various modules and components it consists of. The evaluation studies conducted are explained in Chapter 7 and Chapter 8 concludes by summarizing our work, the benefits and challenges of such an end-to-end framework, and future directions.

Chapter 2

Literature Review

This chapter begins by reviewing the most recent and relevant datasets that are used for model training and benchmarking (Section 2.1). Further sections brief the relevant literature in three directions. Section 2.2 reviews relevant work that has end-to-end pipelines for automatic classification of claims. Section 2.3 explains research related to explainability and interpretability within the fact-verification process and looks into systems that are inclusive of explanations within the framework. Finally, Section 2.4 highlights the importance of the human-in-the-loop (journalists) approach and a journalistic-informed approach to be considered while building end-to-end fake news detection frameworks.

2.1 Relevant Datasets

There are numerous benchmark datasets publicly available and widely used in the related research work, which either is employed for training models or used as benchmarks for evaluating the model performance. This section briefly dives into a few relevant datasets and their characteristics.

LIAR & LIAR-PLUS: With an objective to produce a large dataset to facilitate the development of computational approaches for fake news detection and automatic fact-checking, Wang (2017) presented the LIAR dataset. This dataset contains 12,800 human-labeled short sentences from PolitiFact API from 2007 to 2016, where each statement is evaluated by their editor for its truthfulness. PolitiFact[1], in general, covers a broad range of political topics, where journalists provide detailed justifications with fine-grained labels. The truthfulness ratings are classified into six fine-grained labels: *pants-fire, false, barely-true, half-true, mostly-true, and true*, with a fairly well-balanced distribution. The speakers are a mixture of Democrats and Republicans from US political parties and to ensure better coverage, the data is sampled from various contexts/venues and diverge set of topics. By adding the column 'the extracted justification', Alhindi et al. (2018) extended the LIAR dataset and called it the LIAR-PLUS dataset. This column contains

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

all the sentences from 'Our Ruling' section of the report from PolitiFact and is void of any sentence that has verdict or verdict-related words. Both the datasets are based on real-world news content as they are manually curated by a group of journalists from various online news sources and are checked for accuracy by them.

The benchmarking using LIAR dataset was performed by a Convolutional Neural Network (CNN) model with best filter sizes of (2,3,4), which additionally encoded speaker-related meta-data as well. They showed that the CNN model performed best compared to other models: Majority, Logistic Regression, Support Vector Machines (Crammer and Singer, 2002), Bi-Long Short-Term Memory (Hochreiter and Schmidhuber, 1997), using only text features. Furthermore, improvements on the test data were found when additional meta-data and text were considered as input features.

The LIAR-PLUS dataset is used to enhance the assessment of the truthfulness of a claim by modeling human-provided justification. Alhindi et al. (2018) used feature-based machine learning methods, Logistic Regression (LR) and Support Vector Machines (SVM) with linear kernel and deep learning method, Bi-Directional Long Short-term Memory (BiLSTM) for a binary and a six-way classification problem. The results evidently showed that in both the tasks, the classification F1 score improved when justifications were included.

FEVER: The **F**act **E**xtraction and **VER**ification dataset consists of 185,445 claims that are manually verified against introductory sections of Wikipedia pages (Thorne et al., 2018). The claims are manually generated from Wikipedia by altering them in various ways including changing its meaning. These claims are classified as SUPPORTED, REFUTED, or NOTENOUGHINFO using a verification process that involves a separate annotation process, using suitable guidelines and user interfaces. The construction of this dataset follows a two-step process. The first step is to generate claims using the information extracted from Wikipedia. Sampled introductory section sentences from the 2017 Wikipedia dump, with approximately 50,000 popular pages are given to annotators. They are asked to generate mutated versions of the claims using a single fact. A concept of dictionary is used as additional knowledge to increase the complexity of the generated claim, hence creating a set of extracted and mutated claims. The second step involved labeling these generated claims by the annotators themselves and further find evidence if the claim is labeled as SUPPORTED or REFUTED. The third label is assigned if they could not find any amount of information in Wikipedia. Both these steps had its own guidelines, with 50 annotators (native US English speakers), of which 25 were involved in Step 1. Since the second step has its own complexity, three forms of data validation were conducted: 5-way inter-annotator agreement, agreement against super-annotators, and manual validation by the authors. A simple pipeline approach consisting of document retrieval and textual entailment was developed to evaluate the dataset, which produced a classification accuracy of 31.87%. The major observation about this dataset is that, even though this dataset will serve as baselines for model training for claim verification and related tasks, the data is synthetic in nature. The claims are artificially generated and do not match generated news in the real world.

MultiFC: The dataset introduced by Augenstein et al. (2019), consists of 34,918 claims that are naturally occurring. In contracts to other datasets that are either artificially constructed claims or smaller in size, MultiFC is a collection of claims from 26 fact-checking websites in English along with evidence pages to verify the claims, and the context of their occurrence accompanied by a set of meta-data[2]. The dataset construction is achieved by crawling fact-checking websites in English. The list of fact-checking websites is taken from Duke Reporters' Lab, which resulted in 38 websites, of which ten could not be crawled due to various reasons. The initial crawl led to a collection of 36,534 claims and post a semi-automatic cleansing, the final dataset size is achieved. The crawled information contains the full text along with meta-data like ID, claim label, URL, speaker, checker, and so on. For each claim, text evidence pages are retrieved using Google Search API with top 10 ranks saved.

The task of fact-checking is defined as a Multi-Task Learning (MLT) approach, where each domain is modeled as its own task in MLT architecture, and labels are projected into a fixed-length label embedding space. A formal definition of the MLT tasks, T, are made and the respective training is done on a classic deep learning MLT model. This is considered as the base model, sharing parameters across tasks, which generates a probability distribution for each task using a task-specific softmax output layer. Additionally, to learn the relationship between labels, a label compatibility function measures the label similarity across all tasks. This step is formulated as a Label Embedding Layer (LEL), which operated parallel to an evidence ranking model. Such a modeling is done due to the sheer amount of label variance present in the fact-checking websites. The dot product of claim-evidence embeddings and label embeddings generate the predictions. The experimental setup uses Bidirectional Long Short-Term Memory (Bi-LSTM) model for sentence embedding. Two models, claim-only and evidence-based veracity prediction models are trained and it is shown that the evidence-based veracity prediction outperforms the former by a large margin on the F1 scores.

FakeNewsNet: A set of two comprehensive datasets as a collection of news content, social context, and spatiotemporal information is presented by Shu et al. (2018). The news content is extracted from fact-checking websites such as PolitiFact and GossipCop, where journalists and domain experts provide fact-checking evaluation results for claims, which are used as ground truth. For collecting the social context and user information, APIs from social media like Twitter are used. They provide user engagement details related to fake and real news from fact-checking websites. Search queries created from the headlines of the news articles collect the necessary user information. The spatial information is collected by

[2] http://www.copenlu.com/publication/2019_emnlp_augenstein/

obtaining the locations provided in the user profiles. The respective temporal information is extracted from the timestamps of the respective user engagements. These timestamps also facilitate understanding of fake news propagation on social media and how the topics change over a period of time.

For the fake news detection performance evaluation, Shu et al. (2018) uses the PolitiFact and GossipCop datasets, where individual dimensions (news content, social context & both) are evaluated separately. Social context is evaluated using a variant of the Social Article Fusion (SAF) model, which uses user engagement temporal pattern to identify fake news, known as SAF/A. News content is evaluated using the standard classification methods like Support Vector Machines (SVM), Logistic Regression (LR), Naive Bayes (NB), and Convolutional Neural Networks (CNN), with one-hot encoded vector representation of news articles. News content is also evaluated using the SAF/S model. SAF/S variant utilizes auto-encoders that learn the new article features for classification. For evaluating both news content and social context, the SAF model that combined the SAF/S and SAF/A is used. This model consists of a Long Short-Term Memory (LSTM) with auto-encoder and a second LSTM that captures the temporal patterns of the user engagements. For news content-based methods, SAF/S performs better in terms of accuracy (65.5%), recall, and F1 and SAF/A provides an accuracy of 66.7% whereas, SAF/S has a higher precision when compared to the rest of the baseline models. Another observation is that SAF performs relatively better than SAF/S and SAF/A on both datasets, with an accuracy of 69.1%. The datasets can be accessed using an API, allowing the selection of required datasets. Since this repository provides multiple dimensions of information, it opens up potential applications like fake news detection, evolution over time, mitigation, and malicious account detection.

Discussion: Considering the plethora of datasets available, the most relevant dataset is the LIAR-PLUS and MultiFC, as they are both based on real news content. The evidence-based improvements from Alhindi et al. (2018) shows a good reason to use the LIAR-PLUS datasets. Moreover, it is publicly available and extracted from real-world news. MultiFC would have been a good gold dataset, but have not been used as it is not publicly available yet. The multi-dimensional approach within the FakeNewsNet framework dataset is followed while creating the knowledge base in this thesis (see Section 4).

2.2 Automated Fact-Verification Systems

A very similar system to what we propose here in this thesis is the ClaimBuster, an end-to-end system that uses machine learning, natural language processing, and database query techniques for fact-checking (Hassan et al., 2017). For new claims, the system converts them as queries against various knowledge databases, and for claims that require human intervention, the platform assists using algorithmic and computational tools. The system architecture consists of the following components:

1. **Claim Monitor:** This component continuously monitors sources inclusive of broadcast, social media, and websites. For each source, an extraction method is implemented to collect closed captions, filter politics-related Tweets, and transcripts of proceedings.

2. **Claim Spotter:** Based on the classification and scoring model, the claim spotter discovers factual claims that are worth checking by assigning a score, higher score indicating check-worthy factual claims.

3. **Claim Matcher:** Given the check-worthy claim by the claim spotter, this component searches the curated fact-checked repository and returns those fact-checks matching the claim. The similarity between the check-worthy claim and the facts in the repository are measures using two approaches: 1) Token similarity and 2) Semantic similarity. The results of both these approaches are combined while returning similar fact-checked claims.

4. **Claim Checker:** This component collects supporting or debunking evidence from knowledge bases and the web using the question-answering engine Wolfram Alpha and Google. A possible verdict is also returned if there exists a clear discrepancy between the returned answers and the claim. A *context* is created by grouping the matching sentences and few surrounding sentences. The Wolfram Alpha and Google contexts along with derived verdicts compose the evidence and are presented to the user.

5. **Fact-Check Reporter:** The debunking evidence and the scores from the claim spotter are converted to a report and is delivered to the user through the project website.

A study was conducted by Hassan et al. (2016) using the ClaimBuster system to check the worthiness of the sentences that belong to topics extracted from 21 primary debates of the 2016 U.S. presidential election. The aim was to compare the results of the system to the judgments of professional fact-checkers at CNN and PolitiFact. The resulting observation is that the sentences selected by both CNN and PolitiFact for fact-checking were given scores that were significantly higher by ClaimBuster (and were more check-worthy) than sentences not selected for checking. The relevant dataset that the ClaimBuster is trained on is publicly available[3].

By proposing QABriefs, Fan et al. (2020) introduced a model called QABriefer that performs structure generation through claim-conditioned question generation and open domain question answering. With an objective to generate fact-checking evidence, a set of relevant questions and their answers are generated as fact-checking briefs, known as QABriefs. The three components introduced by them are:

1. Fact-checking briefs: The purpose of this brief is to provide useful evidence to the human fact-checker. Three types of briefs are proposed:

[3] https://zenodo.org/record/3609356#.YLzfDCbhXs0

   (a) *Passage Briefs*, consisting of relevant passages retrieved from Wikipedia, created using pre-trained Dense Passage Retriever.

   (b) *Entity Briefs*, decomposes the claim to smaller entities using BLINK (model trained on Wikipedia data that links entity to its nearest Wikipedia page), retrieves its Wikipedia statements, and provides the first paragraph as the brief.

   (c) *QABriefs*, decomposes the claim into a set of questions and answers, generates briefs for each question.

2. QABriefDataset: This dataset is created to train and evaluate models that generate QABriefs. Existing fact-checking datasets like DATACOMMONS and MultiFC are used to source claims for this dataset. The process involves generating questions from claims, answering these questions based on the source of the claim and the question, and concluded with a validation. It is created using crowdsourcing based on existing fact-checked claims and is a collection of 6,897 QABriefs with 3 Q&A pairs each.

3. QABriefer: This model consists of two fine-tuned BART models, first BART fine-tuned for question generation based on claims and the second BART fine-tuned on QABriefDataset for abstractive answer generation. The QABriefer uses the question generation model to generate multiple questions and for each question, evidence documents are retrieved using a search engine. The questions and the evidence are used by the fine-tuned QA model to generate answers to produce the full brief.

With the hypothesis from the authors, ‘briefs increase the accuracy and efficiency of fact-checking’, they suggest that briefs are a promising avenue for improving crowdsourced fact-checking. Although the complexity of the overall system increases with the accuracy being dependent on the QA models, an F1 score of 32.8 is achieved, when the model is fine-tuned on a large question answering dataset. An observation made is that these briefs introduce biases in the crowd workers, because they submitted fact checks based on the briefs alone and had not used the search bar for additional evidence. They also state “Briefs aid accuracy and efficiency, but are not fully sufficient to produce a verdict”.

A related work that focuses on identifying hoaxes on Facebook based on users that interact with them is presented by Tacchini et al. (2017). Hoaxes as referred to as intentionally crafted fake information and they diffuse rapidly, within the first two hours, in the social network sites. The key idea proposed by the authors is about increasing the classification accuracy by examining the users that interact with hoaxes. For creating the dataset, a set of Facebook pages were selected that either cover scientific topics or deal with conspiracies and fake scientific news, classified under two categories: scientific news sources and conspiracy news sources. Public posts, likes on post, and user information from these selected pages were collected to form *complete dataset*. This dataset comprises of 15,500 posts from 32 pages, 2,300,00+ likes by 900,000+ users. The time period of these posts is from the second half-year of 2016. Two classification methods are used to classify Facebook posts as hoaxes or non-hoaxes taking into account the information of the users that interact with such posts. The first method is formulated as a supervised learning, binary classification problem, where each post is associated with a set of features. Logistic Regression (LR) is employed for classification as the dataset consists of a very large, uniform feature set and has a non-inference property for unrelated users that facilitates learning. The set of features are built based on which users liked which post. The second approach uses a classification algorithm derived from crowdsourcing known as the Boolean Label Crowdsourcing (BLC) problem (de Alfaro et al., 2015). In this classification problem, the posts are labeled as True/False and the BLC problem is to compute the consensus labels from the user input. An adaptation of the standard BLC algorithm is presented by the authors, as the standard BLC assumes that people generally tell the truth. The harmonic algorithm with an adapted learning setting as a set of posts is used for the classification task. This adapted algorithm is used because of its computational efficiency, ability to deal with large datasets, and adaptability to learning sets. Moreover, LR comes with the drawback of not transferring information across users, not in the training set. This drawback is alleviated by propagating information from posts where the ground truth is known, to posts that are connected by common users in the harmonic algorithm.

The experiments were performed to measure two performance characteristics: 1) The accuracy measured as a function of the training set size and 2) The accuracy measured as the amount of information transferred across pages. A cross-validation analysis was conducted on the complete and intersection dataset. For the complete dataset, BLC performs better than LR and for the intersection dataset, the accuracy is vice-versa. Overall, with a dataset of 15,000 posts from 32 pages and approximately 900,000+ users, the cross-validation analysis reports accuracies exceeding 99% for the logistic regression and 99.4% for the harmonic algorithm. Another observation made is since Facebook users naturally revolve around common interests and pages, the classifiers were tested on posts related to pages that they had not seen during the training phase. This corresponds to the second method of measuring accuracy based on the amount of information transferred. Two experiment settings: one-page-out (select only one page in each run as testing data) and half-pages-out (select half pages from the datasets in each run) are used. The results indicated harmonic BLC to be better in transferring information across pages, in both one-page-out and half-page-out experiments.

An approach that combines the advancements in text classification and fact-verification to tackle fake news detection is presented by Li and Zhou (2020). It is crucial to understand the distinction between fact-verification and fake news detection as stated by them: fact-verification aims to check the reliability of a claim of one or a few sentences while fake news detection aims to check the trustworthiness of a long article. The dataset used is an adapted version of the FakeNewsNet (Shu et al. (2018)), which is a benchmark dataset for fake news detection. A manual increase in the ratio of real news is carried out in the test dataset to simulate a real-world scenario, where it is considered that only a small portion of real news is fake. The proposed approach is a combination of well-established pre-trained models for summarization and fact-verification problems. The methodology involves two major steps.

First, the open-source BERT-based extractive summarization model is employed to summarize the input articles into a short claim, with a compression ratio of 0.1, and only the top two sentences predicted are selected. Such a summarization makes the long articles similar to claims available in the FEVER dataset. The reason for selecting only the top two sentences is to minimize the inconsistency in the training and inference process of the fact-verification model. They argue that summarization concentrates the information into a few sentences, hence making the classification easier. In the second step, the generated claim is fed into a pre-trained fact-verification model, GEAR, for veracity classification. GEAR is based on BERT and graph neural network and is trained on the FEVER dataset. As available in the FEVER dataset, this model requires multiple support evidence as input. To satisfy this requirement, a set of support evidence is constructed using the Google search engine. The process involves extracting keywords from the claim using AllenNLP, crawling the web with these keywords using Google, and creating the evidence set using sentence embedding similarity. The similarity is determined using the pre-trained sentence embedding model sentenceBERT. The GEAR model then uses a set of top 5 related sentences as evidence and the input claim to predict the veracity of the claim. GEAR is a three-way classification model, but the third category "not Enough Information" is omitted to make this a binary classification problem. This is achieved by initializing the output layer of the model for a binary classification and using default hyper-parameters while fine-tuning.

For experiments and comparative study, other standard methods like Support Vector Machines (SVM), Logistic Regression (LR), Naive Bayes (NB) are also used as classification models. Under the zero-shot approach, an accuracy of 44.10 and F1 score of 48.32 on PolitiFact and an accuracy of 56.49 and F1 score of 37.42 on GossipCop is achieved, whereas, under the supervised learning approach, an accuracy of 68.75 and F1 score of 72.50 on PolitiFact and an accuracy of 73.74 and F1 score of 52.50 on GossipCop is achieved. The higher accuracies/F1 scores in both approaches compared to its respective standard methods validate the approach of transfer learning from well-trained text summarization and fact-verification model to the task of fake news detection. This approach is also limited by its benefit: using pre-trained models in real-world applications is still computationally expensive.

Discussion: ClaimBuster is very similar to our proposed framework in this thesis from an architecture point of view but differs in the domain of usage. We focus on fake news detection from real-world news whereas Hassan et al. (2017) focuses more on live discourses like interviews, speeches, debates, tweets, and notable political events like the U.S presidential elections 2016, the Australian parliament Hansard. The Briefs presented by Fan et al. (2020) is relevant from the evidence requirement for fact-checking. When a summary of relevant evidence is presented to a fact-checker, it enables informed decision-making with contextual and topic awareness. The presentation of such pieces of evidence as assistive information is added into our framework thereby supporting end-users/journalists to make informed decisions about the truthfulness of the news.

The Facebook hoaxes method highlights the use of user information and their interactions as measures for classification. This work also indicates that with such simple models and the availability of user-interaction information while generating the dataset, we could add additional classification information as domain knowledge to the deep learning model used for veracity prediction. The user interaction information is an additional dimension included in our knowledge base, which opens up a possibility of supportive analysis to the veracity prediction. Finally, the approach proposed by Li and Zhou (2020) is of interest as it enables zero-shot fake news detection without the requirement of a large-scale labeled dataset to train fake news detection models. They focus on transfer learning from two relatively well-studied problems - text classification and fact-verification. A similar transfer learning concept is implemented in our framework, thus reducing the dependency on the availability of large-scale manually annotated data (see Section 6.4 and 6.5).

2.3 Explainability in Fact-Checking

Kotonya and Toni (2020) explains the missing of generating justifications for claim verdicts in fact-verification pipeline and how the same can be modeled along with veracity prediction. According to the surveys conducted by the authors, even though justification of claim verification judgments is arguably a crucial part of manual fact-checking journalistic processes, not much work has gone into acquiring explanations from the existing automated fact-checking systems, either in the form of post-hoc system-generated explanations or manually annotated explanations. By closely examining the trends in various fact-checking methods, there is a strong reliance on training for sub-tasks in the pipeline. This evidently generates a trade-off between system complexity and transparency. An important distinction made is the difference between interpretability and explainability, where the former is to be understood as the ability of a machine learning model to offer ways to analyze and visualize its decision-making process whereas the latter to be understood as the ability of the machine learning model to deliver its decision-making rationale. The emphasis in the presented survey is on increased understanding of reasons of the claim verification using natural language, as human-readable explanations, i.e explainability of the systems. One aspect of explainability that is common in very few existing systems, as observed by them, is that the explanations are extractive in nature. This involves including input components most related to the predictions as part of the explanations. Thereby, the generated explanations are dependent on the input claim itself.

A background on journalistic mechanisms of fact-checking is presented to better understand what these explanations represent in the fact-checking domain. According to Graves (2017), journalists work with a set of guidelines, and the process is categorized into three parts: identification, verification, and correction. Another aspect that is important to understand is the process of gathering these explanations. Various fact-checked sources are available for extracting these explanations and are written by journalists but there exist no unified formats. The survey further focuses on various ways of formulating the explanation generation tasks within the context of automatic fact-checking systems. A few categories, based on the task formulation, mentioned here are attention-based explanations, explanations as rule discovery, explanations as summarization, and generating adversarial claims. The paper also provides a qualitative analysis of explainable fact-checking systems from a perspective of eight desirable properties for explanations: actionable explanations, causal explanations, coherent explanations, context-full explanations, interactive explanations, unbiased or impartial explanations, parsimonious explanations, and chronological explanations. Further, they discuss the current limitations of these systems, such as unverifiable claims, no system providing an explanation for all the sub-tasks in the fact-checking pipeline, and the non-existence of a consistent method for explanation evaluation. Another highlight from this survey is: all existing methods try to explain only one component of the pipeline and the systems as such only explain the predictions.

Another relevant research work in this direction is from Atanasova et al. (2020), which states “producing justification for the veracity prediction - is an understudied problem”. A study on how explanations can be generated by jointly modeling with veracity prediction, based on available claim context is provided. The task is modeled as a summarization task, where the model is provided with elaborate fact-checking reports and the model is required to generate veracity explanations similar to human justifications. By using the PolitiFact-based dataset LIAR-PLUS, this work automatically extracts justifications from the long-ruling comments. Ruling comments, included in the dataset, are summaries of the whole explanation in a few sentences. Two models, explanation extraction and veracity prediction models as well as a third joint model are trained. The models are based on DistilBERT (Sanh et al., 2019), the version which is pre-trained with a language modeling objective and its embeddings fine-tuned for the specific task. The objective of the explanation is to maximize the similarity between the extracted explanation and human justification. The top-4 sentences are selected to achieve the highest ROUGE-2 F1 score (Lin, 2004) when compared to the gold justification. The veracity prediction generated a graded prediction with 6 classes, similar to the classes available in the LIAR dataset. Both the models have DistilBERT as the last layer, which is fed into a feed-forward layer, with sigmoid and softmax activations respectively, and the predictions optimizing cross-entropy loss function. The joint model is similar to the individual models except that the function learned predicts both the veracity explanation and veracity label of the claim simultaneously based on input claim text and its ruling comments. This model optimizes a weighted combination of both losses. The macro F1 still remains under 0.5 for the combined model, given a gold justification and this is assumed to happen because the task itself is challenging or due to the small dataset size. The study presented claims to be the first that generates veracity explanations along with predictions and also shows that veracity prediction can be combined with veracity explanation generation and such a multi-task system improves the overall performance of the veracity system.

Building a text generation framework that can generate responses similar to human fact-checkers is a novel topic in the NLP domain. The latest development in this direction is based on biased TextRank extractive method (Mihalcea and Tarau, 2004) and GPT-2 abstractive method (Radford and Sutskever, 2018) by Kazemi et al. (2021). Biased TextRank varies from the original TextRank by accepting a "bias" as input and ranks the texts using its importance as well as its relevance to the bias term. In the experiments, Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) is used to convert input text into sentence vectors and similarity is computed using cosine similarity. The bias term is used to restart the probabilities in each run of the underlying PageRank (Brin and Page, 1998) algorithm over the text graph, allowing similar nodes to bias query ranked higher. The second method uses fine-tuning pre-trained GPT-2 on a relatively small dataset to generate abstractive explanations using transfer learning. To avoid gibberish/repetitive text, the unimportant texts from biased TextRank are removed from the input during the fine-tuning stage. The dataset used is the LIAR-PLUS and for generating explanations similar to the ruling comments are used. A second dataset from the healthcare domain, Health News Reviews, is used to generate explanations for different evaluative questions. Using automatic evaluation of generated explanations, GPT-2 based model outperforms biased TextRank with a ROUGE-L score of 17.67 when evaluated against actual explanations. The biased TextRank outperforms GPT-2 against the extractive baseline with a ROUGE-L score of 21.88. Additionally, experiments also show that the biased TextRank performs better than the unsupervised TextRank indicating the effectiveness of claim-focused summary for explanation generation. Computationally, biased TextRank is much faster (milliseconds) than the GPT-2 model.

Discussion: As suggested by Kotonya and Toni (2020), for rendering useful and insightful fact-checking explanations, it is crucial to take a journalistic-informed approach, by including them in the framework. This aspect is taken into account while formulating the **Fake News Detection Framework** and is one of the foundations in enhancing the explanation generation component within the framework. From the model presented by Atanasova et al. (2020), it is evident that an explanation can be generated along with veracity prediction as a joint task. They state the future work to be investigating the possibility of explanations by crawling the web for evidence. This step is implemented in our framework, where the evidence is generated by crawling the web (see Section 6.2). Generating explanations can be formulated as a text summarization or a generation task, where both methods have their advantages and disadvantages. We experiment with both these approaches and the final results are presented in Chapter 7. During our experiments, we also have similar observations as Kazemi et al. (2021) in the computational requirement of GPT-2.

2.4 Journalist in the Loop

There always exists a growing demand for efficient human-centered Artificial Intelligence (AI), as stated by Missaoui et al. (2019), to support journalists in researching and verifying information. A co-design workshop conducted with an objective to actively involve journalists in the design activities gave insights about the perspectives of journalists when an automated system is presented to them. An observation made is that journalists generally overestimate the capabilities of automated tools. Even though automated systems can traverse a large amount of data, these systems require the expertise of journalists to determine reliable data sources. A second observation made is the bias and non-transparency issues that arise while using an automated system. As stated in the paper, "They estimate that AI would tarnish the integrity of news by making sources too opaque". This clearly implies that transparency is crucial in the journalism context.

A Hybrid human-AI (HAI) framework for fighting misinformation is presented by Demartini et al. (2020), which leverages the benefits of the following different approaches:

1. The AI scalability in efficiently processing large volumes of data.
2. The fact-checker experts' ability in identifying the truthfulness level of verified statements with transparency and fairness.
3. The crowdsourcing ability in manually processing significantly large datasets.

The framework is envisioned from the limitations of both automated and human-based fact-checking methods. This framework emphasizes the collaboration between humans and AI systems, delivering better transparency on fact-checking processes and allowing end-users to make informed decisions. It consists of the following three main actors:

- Fact-checking experts: They are domain experts who use the other two actors and guarantee that the HAI systems meet three principles: (i) non-partisanship and fairness; (ii) transparency on sources, funding, and methodology; and (iii) open and honest correction policy. This enables optimizing and maintaining high-quality standards of fact-checking process.
- AI methods: The tools that are based on state-of-the-art machine learning algorithms have the ability to deal with large amounts of information/misinformation efficiently produced through different channels. This efficiency can be used by fact-checkers but is not completely error-free as well.
- Crowdsourcing workers: They are in between the above two actors from factors such as cost, scale, accuracy, explainability, and bias control and can be used as an on-demand actor of the framework.

The authors point out that such a hybrid framework has its own benefit of cost-quality trade-offs, load management, and trustworthiness but at the same time, a key question that arises is *who should do what*. The roles that each actor needs to play have to be definitive in nature and they suggest a cascaded model, where actors cooperate to maximize value. An important aspect highlighted is the adaptation of human experts to a hybrid environment requires a certain level of trust in HAI systems. A possible way of achieving this is by embedding AI tools that are self-explainable, guaranteeing transparency at the AI level as well.

Discussion: It is evident from the above literature that an AI-enabled system performs better only when the expertise and domain knowledge from the respective experts are embedded within these systems. An important benefit of including journalists in the loop is that this leads to improved trust in the outcomes of AI system and therefore, can lead to better transparency in veracity prediction. The objective of the framework presented in this thesis is to automate the process of fact-checking by augmenting the knowledge from journalists with meaningful information. Thereby, reducing some of the challenges faced by them while performing fact-checking and creating an assistive framework for their effective decision-making. A second objective is to ensure that the explainability is delivered in human-readable format, which depicts the decision-making process of the framework transparent to the journalists, for effective feedback from them.

2.5 Discussion

In summary, there exist very few works that cover end-to-end pipelines for automated fact-checking. The uniqueness of the framework presented in this thesis is the inclusiveness of journalist expertise and domain knowledge by collaborating with them. As an initial collaboration, a discussion was already achieved with CheckFirst[4], a team that has built a software solution that helps journalists and users to organize and debunk fake news. It is a manual fact-checking system, which is inclusive of a fact-checking framework process from the perspective of journalists. Collaborating and having access to such a framework is key for this thesis as this will be the guidance for building a system that aims at being assistive to journalists. Even though ClaimBuster is a very similar system, it is trained on a specific data set extracted from the U.S 2016 presidential elections. Our system is trained on a knowledge base that is available from the continuous crawl of the relevant news organization and complemented with a check into the existing fact-checking systems. Another important contribution of this work is the integration of generating explanations to support the claim veracity in a human-understandable format. We argue that such a framework is useful for journalists and common users.

[4] https://checkfirst.network/

Chapter 3

Framework's Deep Learning Architectures

Our proposed framework consists of two main tasks where deep learning networks are employed as solutions. The first task is veracity prediction, defined as the prediction of the veracity of the input claim based on given evidence. This task is modeled as a supervised classification problem and we use a deep learning classifier based on a BERT-based transformer model for this classification task. The second task involves explanation generation, defined as the generation of justifications for the veracity of the input claim similar to human-generated justifications. We adapted two well-studied approaches for generating explanations: text summarization and text generation. The T5 model is used for the text summarization task and the text generation task uses the state-of-the-art GPT-2 model. Henceforth, this chapter explains in detail the necessary theoretical background in the respective deep learning architectures used in our framework. The first section describes the general transformer architecture and how it can be adapted to specific tasks. Further, the specific deep learning models, BERT, T5, and GPT-2, which are considered in our framework, are elaborated. Evaluation metrics are crucial to assess the performance of any deep learning model. The metrics used to evaluate our models are explained in the final section.

3.1 Transformers

Basic encoder-decoder architecture is a common paradigm in sequence modeling and transduction problems, which involves any task that transforms an input sequence to an output sequence. The encoder has the function of mapping the input sequences to continuous sequence representations. Further, the decoder generates an output sequence of symbols, one element at a time. Various studies, prominently from Cho et al. (2014), showed that the performance of this basic architecture model deteriorated as the length of the input sequence increased. A variation in this basic structure was introduced by Bahdanau et al. (2015) to address the machine translation problem with better efficiency. An extension allowing the encoder-decoder model to align and translate jointly achieved significant improvements in translation performance. This alignment encodes the input to a sequence of vectors and allows selecting a subset of these sequences adaptively while decoding. This approach is known as *attention* mechanism in the neural transduction models. Parikh et al. (2016) extended this approach by adding intra-sentence attention, which encodes compositional relationships between words in each sentence. This mechanism is known as self-attention and is an important component in the transformer architecture by Vaswani et al. (2017).

With the introduction of transformers in 2017 by Vaswani et al. (2017), they have become the core approach in language understanding tasks like language modeling, machine translation, and question answering. Transformers follow the general encoder-decoder architecture and are enhanced with self-attention and fully connected layers in both the encoder and decoder components. Figure 3.1[1] illustrates the full-scale architecture of the transformer with multi-head attention and scaled dot-product attention layers.

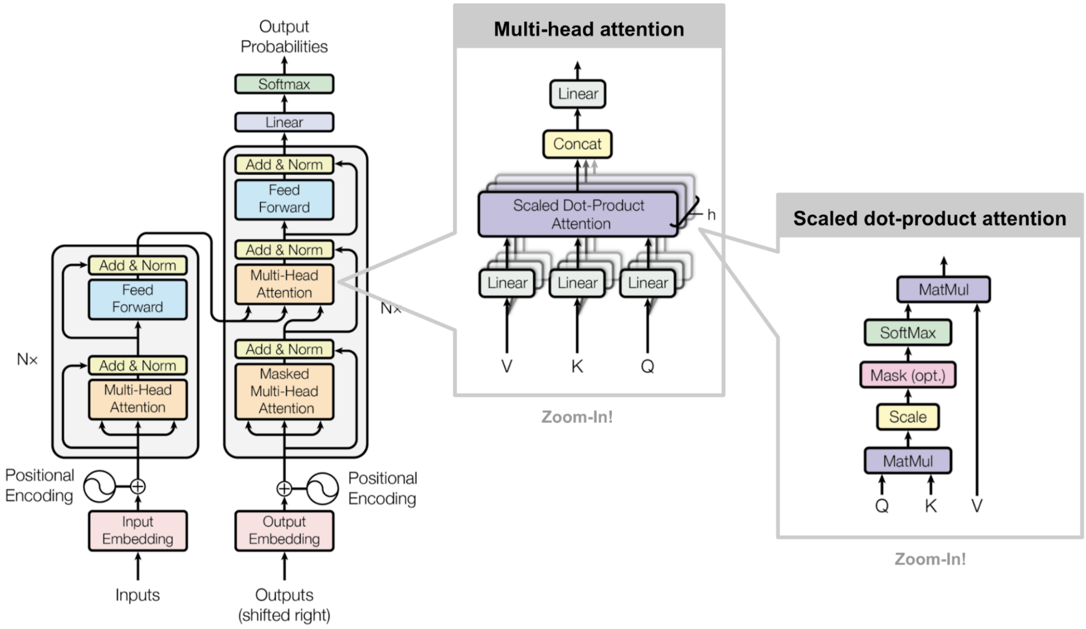


Figure 3.1: The Transformer model architecture (Vaswani et al., 2017)

Attention Mechanism

An important concept within the transformer architecture is the attention mechanism. Intuitively, it means that we need to either attend/omit parts of the input as they influence the later output information. More formally, attention is a means of selectively weighing different elements in the input to have an adjusted impact on the hidden states in the following layers. The attention function as defined

[1] https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

by Vaswani et al. (2017) is mapping a query and a set of key-value pairs to an output, where the queries, keys, values, and outputs are all vectors. The output is a weighted sum of values, where a compatibility function computes each weight using the query and its respective key. The dot product of the key and query provides the attention weight, which is squashed using a softmax function across all attention weights so that the total weights sums to one. This particular attention implementation is referred to as "Scaled Dot-Product Attention". The value vectors corresponding to each input element are then summed according to their attention weights before being fed into subsequent layers.

A second way of applying attention is known as "Multi-Head Attention", where the queries, keys, and values are projected linearly with different learned linear projections to their respective dimensions. Further, on each of these projections, an attention function is applied in parallel, yielding the respective output values. Multi-head attention allows the model to jointly attend to information from different representation sub-spaces at different positions.

It is important to understand how the transformer architecture uses this multi-head attention in its components:

- Like in a typical encoder-decoder attention mechanism in sequence-to-sequence models, the queries are from previous decoder layers, and (key, value) pairs are from encoder output. This allows every position in the decoder to take into account all positions in the input sequence, in the "encoder-decoder attention" layers.
- Self-attention layers are contained in the encoders, where keys, values, queries are from the same place, i.e, the output of the previous layer encoder.
- The auto-regressive property is preserved by the scaled dot-product attention that masks out all input values of the softmax, which correspond to illegal connections.

Point-wise Feed Forward Networks

The feed-forward layer is composed of two linear layers with a Rectified Linear Unit (ReLU) in between them (see Figure 3.1). That is, the input is first transformed by a linear layer (matrix multiplication), the resulting values are then clipped to be always 0 or greater, and finally, the result is fed into a second linear layer to produce the feed-forward layer output. Each sub-layer of the encoder and decoder is connected to this fully connected feed-forward network along with ReLU activation.

Transformer Architecture: The complete transformer architecture is summarized as follows:

- Encoder-decoder architecture, similar to the sequence transduction models.
- The inputs are converted to tokens using learned embeddings.

- A positional encoding using sine and cosine functions of different frequencies to inject sequence order information.
- Encoder and decoder are a stack on N identical layers, respectively.
- Encoder and decoder layers each consist of multi-head self-attention blocks, followed by a feed-forward layer and augmented by residual connections.
- A final softmax classifier that generates the next-token pseudo-probability distribution on the output vocabulary set.

The transformer present by Vaswani et al. (2017) stacks six encoder layers and six decoder layers, with two sub-layers in each layer respectively with 65M learnable parameters.

3.2 Why Transformers & not RNNs/CNNs

Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) models are quite identical in their core properties, where it relies on sequence processing, sentences processed word by word, and assumption of the Markov model, causing the hidden states to retain the past information. The 'one word at a time' concept makes these networks perform poorly as the length of the input sequence increases. Sequential processing renders them not suitable for parallel processing and the requirement of a stack of kernels in CNNs increases the computational complexity.

Transformers are the preferred choice over CNNs and RNNs for the same reason of being non-sequential in nature, its self-attention mechanism (multi-head attention), and positional embeddings both providing information about the relationship between words. The absence of recurrent or convolutional layers, allows them to be trained significantly faster.

From computational complexity, the self-attention layer connects all positions with a constant number of sequential executions whereas RNNs requires $O(n)$ sequential operations, whereas CNNs require $O(n/k)$ convolution layer stacks or $O(log_k(n))$ stacks in dilated convolutions. In essence, transformers are better than other architectures because they avoid recursion, by processing sentences as a whole and learning relationships between words using multi-head attention and positional embeddings. There still exists a drawback for this architecture: the dependencies are captured only within the fixed input size used while training them.

3.3 Bidirectional Encoder Representations from Transformers

A simple transformer architecture limits itself by being able to attend only to its previous tokens in the self-attention layers. This limitation can restrict the contextual learning that is required in downstream tasks like question and answering, language inference. Devlin et al. (2019) introduced a variation in the baseline transformer architecture by enabling it to learn from both left and right contexts while encoding the input sequences. This variant is known as the Bidirectional Encoder Representations from Transformers (BERT) model. The unidirectional constraint is alleviated by introducing the "Masked Language Model" in the pre-training of the BERT architecture. The model is also pre-trained for a second task known as the "Next Sentence Prediction" using a large corpus. The following explains the architectural additions that are made to the vanilla transformer model.

Architecture: The architecture of the BERT model is based on the original implementation of transformers described in Section 3.1, with stacked layers of bidirectional encoders. Two models $BERT_{BASE}$ and $BERT_{LARGE}$ that vary in their number of layers ($L = 12/24$), hidden sizes ($H = 768/1024$) and number of self-attention heads ($A = 12/16$), with total parameters of 110M and 340M, respectively are pre-trained. The input and output representations known as "sequence" is either a single sentence or a pair of sentences packed together, where a sentence can be an arbitrary contiguous text span. WordPiece embeddings with a 30,000 token vocabulary are used, along with special tokens [CLS], [SEP] indicating the first token of the sentence and separation between sentences, respectively.

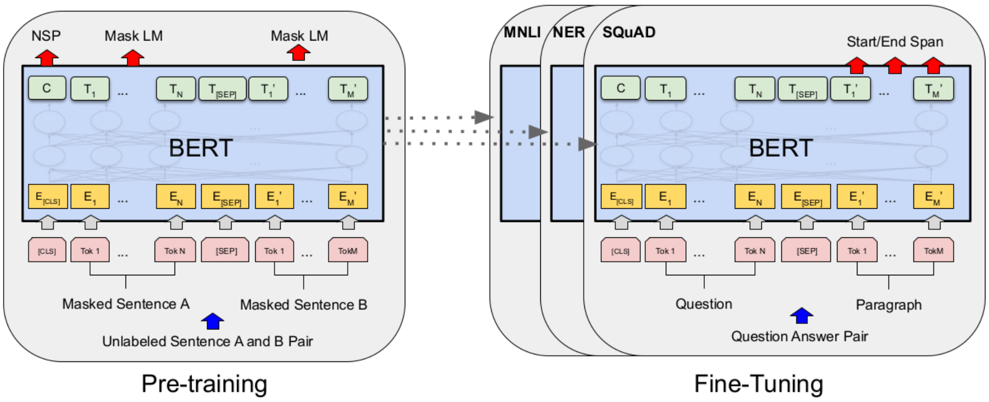


Figure 3.2: Bidirectional Encoder Representation from Transformers (BERT) pre-training and fine-tuning architectures (Devlin et al., 2019)

Pre-Training: Pre-training of the BERT model is achieved by specifying two unsupervised tasks. Task 1 is Masked Language Model (MLM), where a certain percentage of the input tokens are masked randomly and the task is to predict those masked tokens. In the presented experiments, 15% of all WordPiece tokens in each sequence are masked at random and correspondingly the final hidden vectors are fed into an output softmax over the vocabulary. To mitigate the mismatch between pre-training and fine-tuning, 80% of the time, words are masked with [MASK] token, 10% with a random token, and rest 10% with the unchanged original token itself. The final vector representation T (see Figure 3.2) thus generated is used to predict the original token with cross-entropy loss. Task 2 involves learning sentence relationships, called the Next Sentence Prediction (NSP) task, where while choosing two sentences, 50% of the time, the actual sentence is paired and the rest of the time, a random sentence is paired from a monolingual corpus. It is important to note that BERT is an encoder-only model, in the sense that its pre-training goal is to only reconstruct the masked tokens as encoder output. The pre-training corpora used are BooksCorpus (Zhu et al., 2015) and English Wikipedia with 800M and 2500M words, respectively.

The uniqueness of BERT lies in its ability to alleviate the unidirectional language constraint by using the MLM pre-training objective and self-attention mechanism fusing left and right context from an input sentence. This allows fine-tuning BERT by simply plugging in the task-specific inputs/outputs. The authors fine-tuned this pre-trained BERT model on eleven NLP tasks and obtained new state-of-the-art results.

3.4 Text-to-Text Transfer Transformer

By formulating every text processing problem as a "text-to-text" unified redefinition, Raffel et al. (2019) presented the T5 model. Such a unifying approach allows applying the same model, objective, training procedure, and decoding process to every task at hand. T5 refers to **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer and is based on the vanilla transformer architecture.

Architecture: The T5 model remains largely equivalent to the original transformer model along with the following modifications:

- The normalization bias layer is removed.
- Normalization layer moved to the outside of the residual path.
- A different position embedding scheme is used.

Different model architectures are formed by using distinct masking patterns in the self-attention mechanism. The first model is the $BERT_{BASE}$ model, "encoder-decoder transformer", tweaked by stacking two layers instead of one, resulting in 220M parameters. The encoder uses a "fully-visible" attention mask, where the self-attention allows to attend to any input entry when producing output entry. This change in masking pattern results in a "prefix Language Model (LM)", which has the ability to provide the model with prefix/context before making predictions in the text-to-text framework. When compared to the MLM in BERT, which masks 15% of the tokens, the prefix LM splits a text span into two components: 1) input to the encoder and 2) target sequence to be predicted by the encoder without any masking. The decoder uses a "casual" masking pattern, which prevents the

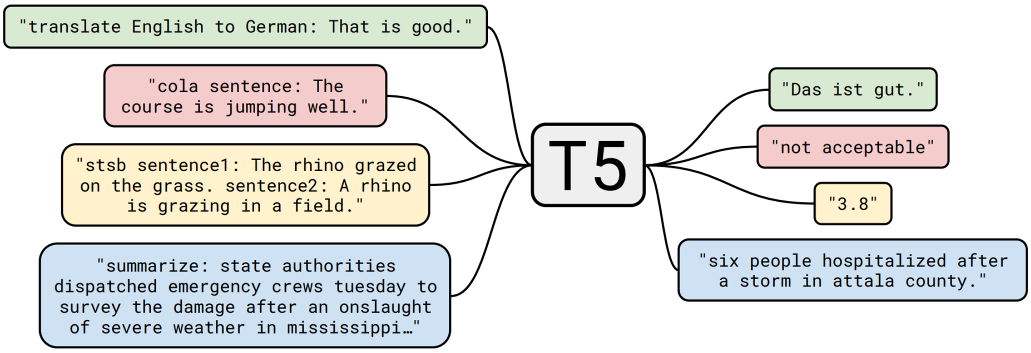

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 3.3: Text-to-Text Transfer Transformer (T5) framework (Raffel et al., 2019)

model from attending to the future entries in the input sequence while producing the output sequence. The authors state that this masking pattern is used while training so that the model cannot "see into the future" as it produces its outputs.

The second model consists of using a transformer decoder (without encoder) as a language model. The objective of such a model is only to predict the next step. The decoder uses the "casual" masking pattern and this model is referred to as the "Language Model (LM)". The casual masking pattern enforces the decoder output to be dependent on the entries up until now. In a text-to-text framework, this dependency is cited as a drawback as it limits the model to only attend the prefix representations. This issue is mitigated by replacing the masking pattern with the "fully-visible" masking during the prefix portion of the sequence. This architecture is the third model and is known as the "prefix LM". This architecture is similar to an encoder-decoder model with the attention replaced with full attention across input and target sequence.

Pre-Training: A teacher forcing technique where both input and target sequence are always needed is employed for pre-training T5. The dataset used is a cleaned version of the Common Crawl[2] web dump, which is two magnitude orders larger than Wikipedia, known as C4 (Colossal Clean Crawled Corpus). The cleaning process involved deduplication, discarding incomplete sentences, and removing noisy content. Three pre-training strategies with a different objective for each are carried out.

1. Prefix language modeling objective: Here a span of text is split into two components, one as input and the second target sequence for the decoder to predict. This is the auto-regressive style language modeling objective

2. BERT style masked language objective: Where 15% tokens are masked, of which 90% are replaced with a special token and the rest with a random

[2] http://commoncrawl.org/

token. The decoder uses sequence as the target without masking during training.

3. Deshuffling denoising objective: The approach involves shuffling a sequence of tokens to use as input and use the original deshuffled sequence as the target.

All the strategies involved training using the standard maximum likelihood method teacher forcing, a cross-entropy loss function, and AdaFactor optimization (Shazeer and Stern, 2018). Each model is pre-trained for 2^{19} steps, with maximum sequence length 512 and batch size 128 using the C4 dataset. Multiple sentences are packed to form standard 2^{16} tokens across all batches. The learning rate used is a generic inverse square root schedule. The authors fine-tuned T5 pre-trained models for various English-based NLP problems such as document summarization, sentiment classification, question answering to compare the effectiveness of different transfer learning objectives.

3.5 Generative Pre-Trained Transformer

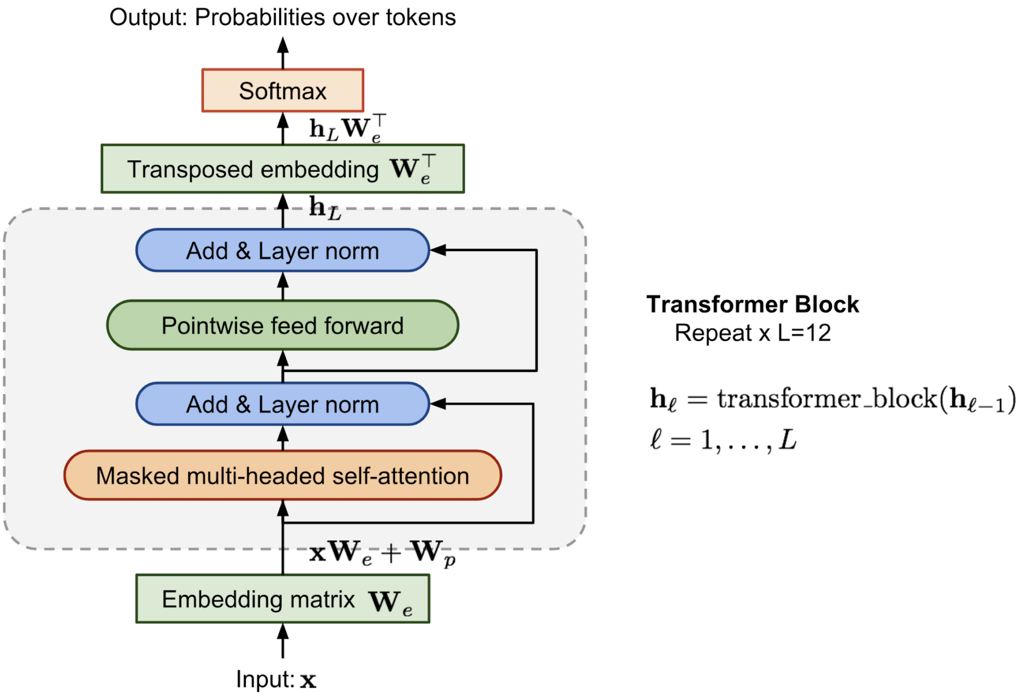
The Generative Pre-Trained Transformer (GPT) from OpenAI[3] is a large scale transformer-based language model, with 1.5 billion parameters, which is pre-trained on a large text corpus. This pre-training allows the model to be competitive in multi-lingual and multiple task domains rendering it useful for the language generation task (Radford and Sutskever, 2018). A major difference between GPT and BERT is that GPT is built by stacking transformer decoder blocks while BERT is built on transformer encoder blocks.

Architecture: GPT-2 is built as a stack of transformer decoders with 12 layers of transformers, each with 12 independent attention mechanisms called the "head". This allows for 144 distinct attention patterns, which allows the model to capture linguistic regularities. The model is mostly similar to the vanilla transformer with masked self-attention heads (768-dimensional states).

Pre-training: The training objective of GPT-2 is purely to predict the next word, given all the previous words within the text. The model is trained on general data, which facilitates fine-tuning for specific tasks with the appropriate data.

The training process followed by Radford and Sutskever (2018) involves two stages. The first involves learning a high-capacity language model using a large corpus text and the second is fine-tuning for a discriminative task. For learning the language model, an unsupervised corpus of tokens is presented to the model to maximize the likelihood. The model applies a multi-headed self-attention followed by point-wise feed-forward layers on the input tokens to produce a distribution over the output tokens. This architecture is very similar to BERT, except that this

[3] https://openai.com/blog/better-language-models/



Output: Probabilities over tokens
Softmax
$\mathbf{h}_L \mathbf{W}_e^\top$
Transposed embedding $\mathbf{W}_e^\top$
$\mathbf{h}_L$
Add & Layer norm
Pointwise feed forward
Add & Layer norm
Masked multi-headed self-attention
$\mathbf{x}\mathbf{W}_e + \mathbf{W}_p$
Embedding matrix $\mathbf{W}_e$
Input: $\mathbf{x}$
Transformer Block
Repeat x L=12
$\mathbf{h}_\ell = \text{transformer\_block}(\mathbf{h}_{\ell-1})$
$\ell = 1, \ldots, L$

Figure 3.4: Decoder block in GPT-2 (Radford and Sutskever, 2018)

unsupervised learning operation is performed over a stack of encoders in BERT, whereas in GPT-2, it is performed on decoders.

Discussion: The encoder-decoder structure is been used for a long time in the NLP problems and hence we use different variants of the base transformer architecture as our models. Each model is fine-tuned using the task-specific datasets as per the task definition and further used in the **Fake News Detection Framework**. The following section explains the evaluation metrics used to measure the performance of the various models.

3.6 Evaluation Metrics

The evaluation metrics are chosen according to the task at hand. For quantitative evaluation, the automatic evaluation scores such as accuracy, F1, and ROUGE scores are measured and for qualitative evaluation, the test data is manually scanned to analyze the model outputs.

Quantitative Evaluation Measures

Loss Visualization

Loss visualization is considered one of the most useful approaches in analyzing the network performance by intuitive interpretation. This method, proposed by

Goodfellow and Vinyals (2015), allows visualizing the training and convergence behavior of the model. We visualize the loss during the training to understand when the model has converged by observing the flatness of the curve. The loss function that we use while training is the cross-entropy function and the objective of all trainings is to minimize this loss. This function, based on entropy, measures the performance of the model on its deviation from the actual label. This is a typical measure used in a supervised classification problem.

Accuracy

Accuracy usually refers to the classification accuracy and we use this to measure the ratio of the number of correct predictions to the total number of input samples in our veracity classifier model. Accuracy is a good measure to consider when there are samples in the dataset that are well balanced between the classes.

F1 Measure

The most common measure used in quantitative analysis of neural network models is the F1 score, which is defined as the harmonic mean between precision and recall. Precision refers to the ratio of the number of correct positive results to the number of correct positive results predicted by the classifier. On the other hand, recall is the number of correct positive results divided by number of all samples. By determining the harmonic mean between precision and recall, the F1 score tells us how precise and robust the classifier is. The score lies between [0,1] and higher the score, the better is the performance of the model.

ROUGE Score

A measure that is used to determine the similarity between a candidate document and a collection of reference documents is the **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation (ROUGE) score (Lin, 2004). A document in this context is referred to as a collection of texts. We use this score to automatically and quantitatively evaluate the summaries produced by the summarization models used on our explanation generation component within the framework. ROUGE-1 and ROUGE-2 look at uni-gram and bi-gram occurrences in both the documents and ROUGE-L determines the longest common sub-sequence existing in both the documents. This measure is an F-score measure hence, it determines precision and recall. In the context of text summarization, an example ROUGE-n score of 40% means that 40% of n-grams in the reference summary are also present in the generated summary.

Qualitative Evaluation Measures

To better understand the performance of the models used in our framework, we also manually examined samples by randomly selecting them from the test dataset.

For the veracity classifier, it is useful to visit the misclassified claims based on the input evidence allowing us to understand the dependency of performance of the model based on the training dataset (see Section 5.1). For summarization and text generation models in the veracity explanation, we manually examine the texts to understand the semantics in reference to the original claim text, its article content, and related evidence.

3.7 Discussion

The transformer models used in our framework fall into three categories: auto-encoding models, sequence-to-sequences models, and auto-regressive models.

Auto-encoding models are based on pre-training models by masking input tokens and reconstructing the original sentences using predicted tokens. The models build a bidirectional representation of the entire sentence and are apt for tasks that involve sentence or token classification. BERT is one such model that is pre-trained by jointly conditioning on both left and right contexts. We use BERT as our veracity classifier model for predicting the truthfulness of an input claim. Sequence-to-Sequence (seq2seq) tasks involve transforming an input sequence to an output sequence and typically cover tasks like translations, summarization, and question answering. The models involved in these tasks use both the encoder and decoder of the original transformer. We use the T5 model, which encompasses a unified framework that converts seq2seq tasks into a text-to-text format. This model is employed in the justification generator summarization approach within our framework.

An auto-regressive model is pre-trained for the classic language modeling task, which involves predicting the next token having seen all the previous ones. The decoder in the vanilla transformer model corresponds to this task modeling by using masks on full sentences, so that the attention heads can only see what is needed. Such pre-trained models are apt for applications like text generation. GPT-2 is an auto-regressive model and we use it in the justification generator text generation approach.

Furthermore, these transformer models consist of minimal task-specific parameters, which allow efficient fine-tuning for downstream tasks using domain-specific datasets. All these models are based on transfer learning, where a model is pre-trained on a data-rich task and then fine-tuned as per the specific task requirement. This learning approach makes it apt for using these models in our framework as this allows us to take advantage of using a pre-trained model that already contains a massive amount of compressed knowledge and mitigates the need for massive datasets for training the models. It is also known that training these models from scratch is computationally heavy and hence, using pre-trained models for fine-tuning is more feasible.

Chapter 4

Knowledge Base

The first step while formulating the framework is to create a knowledge base that allows the extraction of the required datasets for the respective model training. In this chapter, we explain in detail the methodology of creating the knowledge base, its features, and analysis.

4.1 Comprehensive Knowledge Base

In this thesis, we introduce an ever-growing Knowledge Base (KB), known as Comprehensive, Multi-Dimensional Knowledge Base (CompKB) (see Section 4.4). The objective of creating such a KB is to continuously monitor online news sources and create persisted collections of news articles post-processing them. This KB consisting of naturally occurring claims from the internet, in the English language and is realized using a crawler architecture as shown in Figure 4.1.

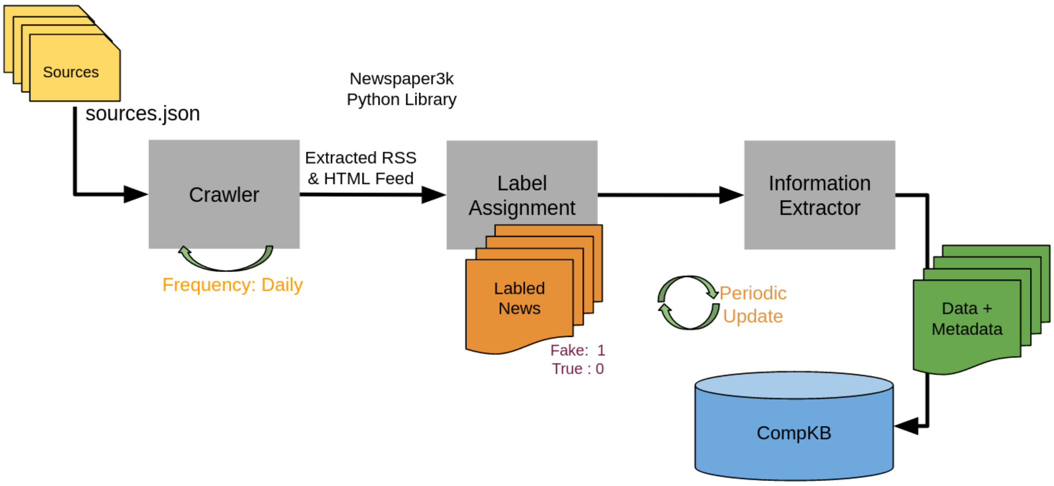


Figure 4.1: Crawler architecture for creating and appending CompKB

4.2 Sources

For creating this knowledge base from the news articles, we consider the following news websites:

- Real News: CNN, BBC, The Guardian, Fox News, NBC News, Washington Post, Aljazeera, DW & France24
- BreitBart, The Onion & InfoWars

These news organizations are largely recognized as sources of real and controversial news, respectively. A few sources like Aljazeera, DW, and France24 are considered here due to the availability of multi-lingual articles, which could be extracted in the future. This is a non-exhaustive list, and they are available as a JSON file (sources.json), where additional sources can be appended, to achieve broader coverage in the future. The respective JSON files contain both HTML and RSS feed links of the respective news website.

4.3 Approach

CompKB is a growing repository, ever since its creation. A web crawler component is continuously scraping articles from the above-mentioned sources. In creating the CompKB, we start by scraping articles from the sources mentioned above and only those articles that have valid published dates are extracted. The HTML, as well as the RSS feeds from these sources are crawled to ensure collection of all possible articles from the respective websites (see Figure 4.1). An upper bound of 100 articles per day per source is set. The extraction is achieved using the Python 3 Newspaper3K library[1], build on top of requests for parsing LXML, allowing to parse articles from URLs. For each source, using a single parse of the URL, the title, text, keywords, and published date are extracted from each article. For each article parsed, the next step is label assignment. Depending on the source from which the article is scraped, it is classified as fake or real. This label assigned set of articles is passed through an information extractor component, where each article along with its meta-data is stored as formatted Python DataFrames. This component is required to convert the raw data that the crawler retrieves into a format that is suitable for further analysis. These DataFrames are committed to the CompKB as unstructured data in tab-separated values formatted files. The crawler runs daily to collect articles that are back-dated by a day from the previous date and the information extractor is run periodically to commit the data to CompKB. Even though an upper bound of 100 articles per day per source is defined, there are only a lesser number of articles scraped, as evident from plot in Figure 4.2.

The plot in Figure 4.2 is a typical distribution of downloaded articles for a span of four days (23.02.2021 to 27.02.2021). The information extractor periodically

[1] `https://newspaper.readthedocs.io/en/latest/`

curates the necessary data from daily scraped news articles and commits the same in the required format to the CompKB.

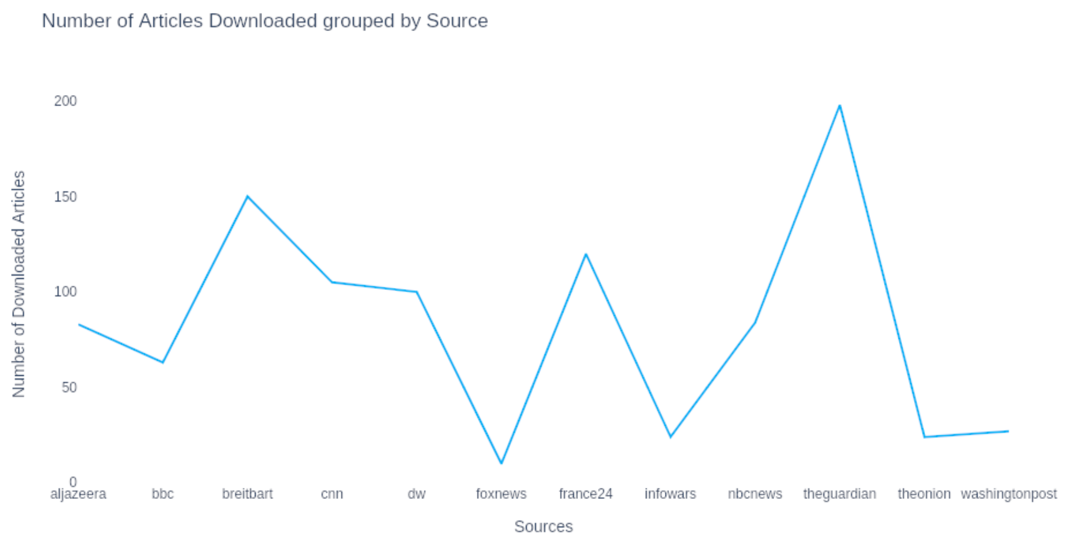


Figure 4.2: Source-based distribution of scraped articles

4.4 CompKB Features

The major contribution of the Comprehensive Knowledge Base is that it is continuously growing and also allows journalist feedbacks to be integrated. Using a feedback process, the KB can be enhanced at any point in time. The feedback method is explained in Section 6.6.

The second contribution is that the CompKB encompasses multi-dimensional information. The following are the information that is being continuously captured as part of this growing KB:

- Temporal Information: The articles saved to the CompKB are those only with a published date. This is intentionally done to ensure that a timeline within the KB is maintained.

- User Information: The meta-data contains the author information of the article, which will facilitate further user-based investigation as required in the future.

- Source Information: It is crucial to also maintain the URL information as this could be used later to cross-check if these articles have been removed in the future, which could act as an indication of the article containing a controversial topic.

This knowledge base could form the baseline for further research directions like determining the trends of fake news topics, or as time advances, which topics dominate the fake news as well as how long they propagate in the online network. The user information could be a piece of potential information for finding more about the fakeness of the article by building the user profile by tracking their related social media engagements, similar to the approach mentioned in Shu et al. (2018). Using this KB as the baseline, we extract the dataset for our veracity classifier model. The next chapter explains this in detail.

Chapter 5

Datasets

Our architecture for the **Fake News Detection Framework** consists of three deep learning models (see Section 6). Each model is fine-tuned with a dataset that is specifically created for the task at hand. In this chapter, we explain the datasets used for fine-tuning the deep learning models and analyze these datasets to understand their features.

5.1 TrueFake Dataset

The first dataset curated is for the **Veracity Classifier** model, and is a subset of the **CompKB**, explained in the previous section. A subset of 1,313 articles, scraped over the month of January 2021 was extracted. A second source, fake and real news dataset by Ahmed et al. (2018, 2017), where the real news was collected from reuters.com and fake news was collected from websites that is been pointed by PolitiFact as fake, is appended to the scraped collection of articles. This publicly available dataset is manually verified during the creation process, as stated by the authors, and has a good balance between fake and real news claims. Thus curated data is then used as the baseline dataset for training the **Veracity Classifier** model.

A total of 46194 claims with their meta-data is the resulting dataset, where 22,365 (48.42%) claims belong to the true label and 23,829 (51.58%) claims belong to the fake label. The time period of the final dataset is shown in Figure 5.1, the peak indicates the latest scraped articles added from the web crawler component, at the time of creating the subset of the CompKB. Appending the scraped (1,313) articles from CompKB adds noise to the fake and real news dataset by Ahmed et al. (2018, 2017) and is the gold dataset for the **Veracity Classifier** model. This dataset is referred to as TrueFake Dataset. 60% of the dataset is used for training the model, 20% as validation, and the rest as the test set. It is important to note that, the test subset consists only of the data from the fake and real news dataset, without the noise from scraped articles. The snippet of the dataset is shown in Figure 5.2

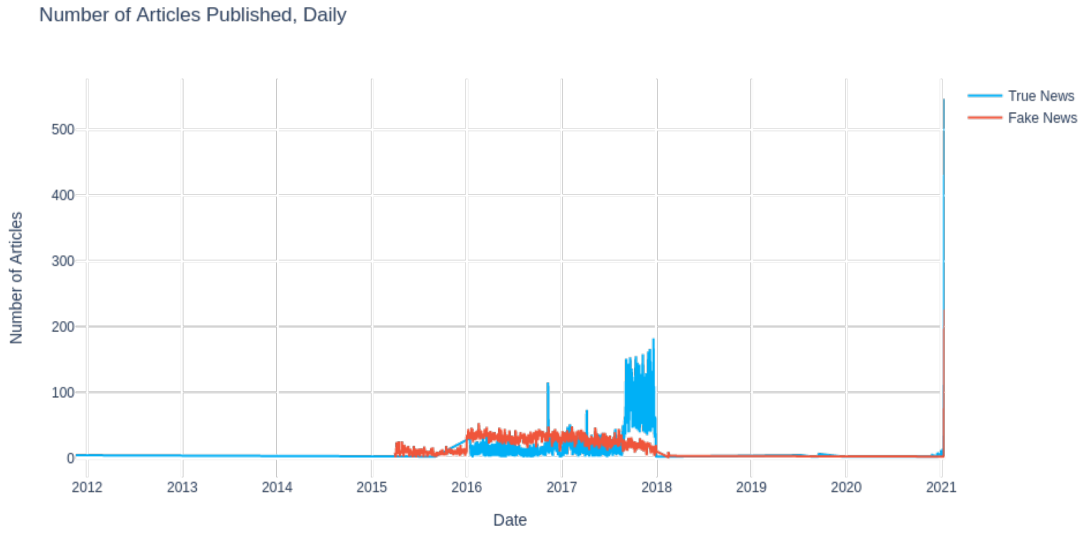
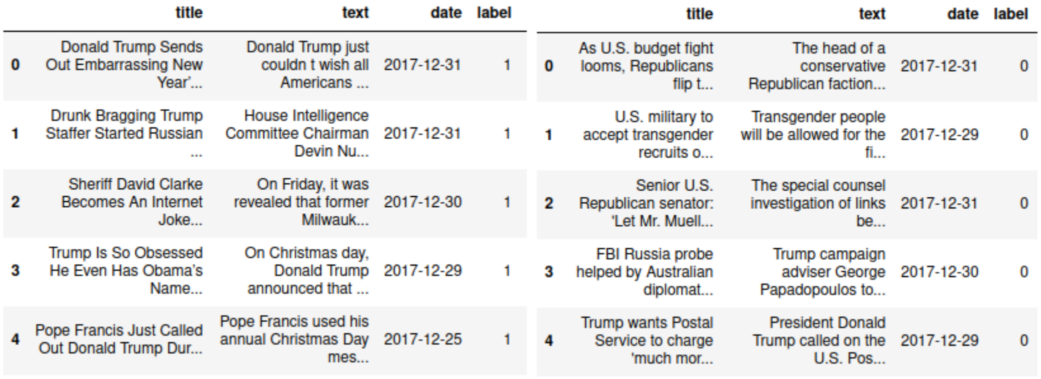


Figure 5.1: Timeline distribution of the TrueFake dataset



	title	text	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	2017-12-31	1
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	2017-12-31	1
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	2017-12-30	1
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	2017-12-29	1
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	2017-12-25	1

	title	text	date	label
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction...	2017-12-31	0
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fi...	2017-12-29	0
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links be...	2017-12-31	0
3	FBI Russia probe helped by Australian diplomat...	Trump campaign adviser George Papadopoulos to...	2017-12-30	0
4	Trump wants Postal Service to charge 'much mor...	President Donald Trump called on the U.S. Pos...	2017-12-29	0

Figure 5.2: Snippet of true and fake articles in TrueFake dataset

5.2 TrueFake Dataset Analysis

We conducted a detailed study of the curated TrueFake dataset to get better insights into the data. The first observation made is that real news contains a lesser average number of words than fake news. The same applied to the headlines of these articles, where the fake news had longer average number of words than the real news (see Figure 5.3). We also looked into the sentiment of the headlines and found that there is not much difference in the sentiments for fake and real news. The plots in Appendix B indicate the evidence of these observations.

The average number of words in a real news article is 394 and the fake news article is 424, with a higher standard deviation (407.56) and more articles on the right tail for fake news compared to the standard deviation (301.51) of real news.

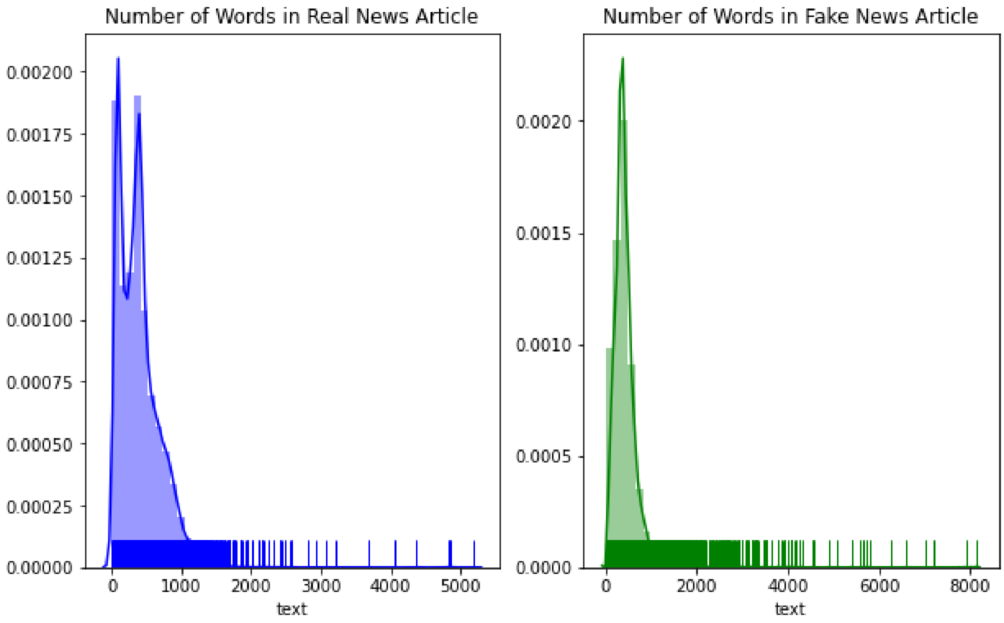
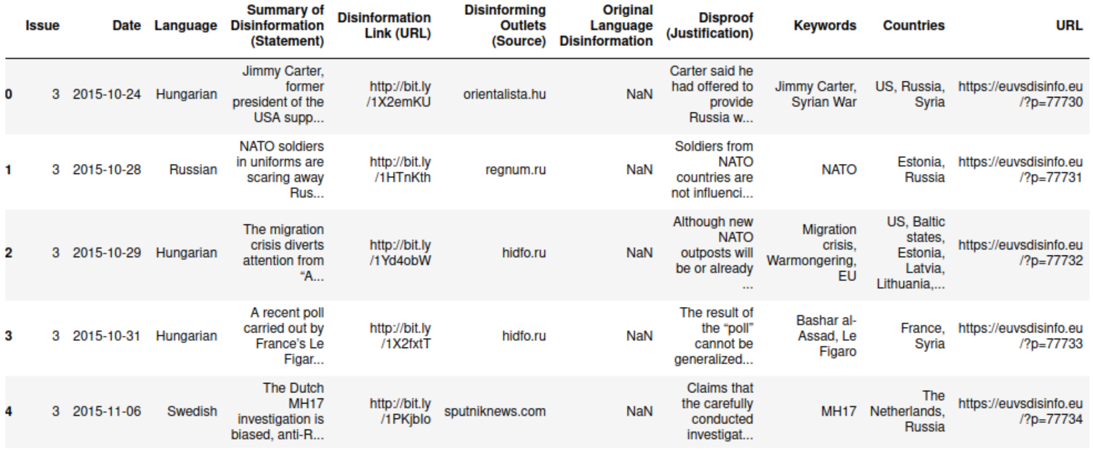


Figure 5.3: Number of words in real vs fake news articles in TrueFake dataset

5.3 EUvsDisInfo Dataset

CLARIN (Common Language Resources and Technology Infrastructure) organized a virtual hackathon on the detection of disinformation in the context of the COVID-19 pandemic. It was held virtually between 21 September - 15 October 2020, to bring cross-disciplinary groups of researchers to work on the task of disinformation detection[1].



	Issue	Date	Language	Summary of Disinformation (Statement)	Disinformation Link (URL)	Disinforming Outlets (Source)	Original Language Disinformation	Disproof (Justification)	Keywords	Countries	URL
0	3	2015-10-24	Hungarian	Jimmy Carter, former president of the USA supp...	http://bit.ly /1X2emKU	orientalista.hu	NaN	Carter said he had offered to provide Russia w...	Jimmy Carter, Syrian War	US, Russia, Syria	https://euvsdisinfo.eu /?p=77730
1	3	2015-10-28	Russian	NATO soldiers in uniforms are scaring away Rus...	http://bit.ly /1HTnKth	regnum.ru	NaN	Soldiers from NATO countries are not influenci...	NATO	Estonia, Russia	https://euvsdisinfo.eu /?p=77731
2	3	2015-10-29	Hungarian	The migration crisis diverts attention from "A...	http://bit.ly /1Yd4obW	hidfo.ru	NaN	Although new NATO outposts will be or already ...	Migration crisis, Warmongering, EU	US, Baltic states, Estonia, Latvia, Lithuania,...	https://euvsdisinfo.eu /?p=77732
3	3	2015-10-31	Hungarian	A recent poll carried out by France's Le Figar...	http://bit.ly /1X2fxtT	hidfo.ru	NaN	The result of the "poll" cannot be generalized...	Bashar al-Assad, Le Figaro	France, Syria	https://euvsdisinfo.eu /?p=77733
4	3	2015-11-06	Swedish	The Dutch MH17 investigation is biased, anti-R...	http://bit.ly /1PKjblo	sputniknews.com	NaN	Claims that the carefully conducted investigat...	MH17	The Netherlands, Russia	https://euvsdisinfo.eu /?p=77734

Figure 5.4: Snippet of EUvsDisInfo dataset

[1] https://www.clarin.eu/event/2020/hackathon-covid-19-related-disinformation

The EUvsDisInfo dataset was manually curated by one of the dataset presenters, Madis Vaikmaa[2]. He is a Journalist in the EUvsDisInfo project, which is a flagship project of the European External Action Service's East StratCom Task Force. The dataset curated by him consists of 9551 entries that correspond to the Russian Federation's ongoing disinformation campaigns. Each entry in the dataset is a disinformation and contains human-annotated justification as to why it is a disinformation along with the web link and justification in the original language. The snippet of the dataset is shown in Figure 5.4 and the data contains disinformation between 2015 and 2020 (see Figure 5.5).

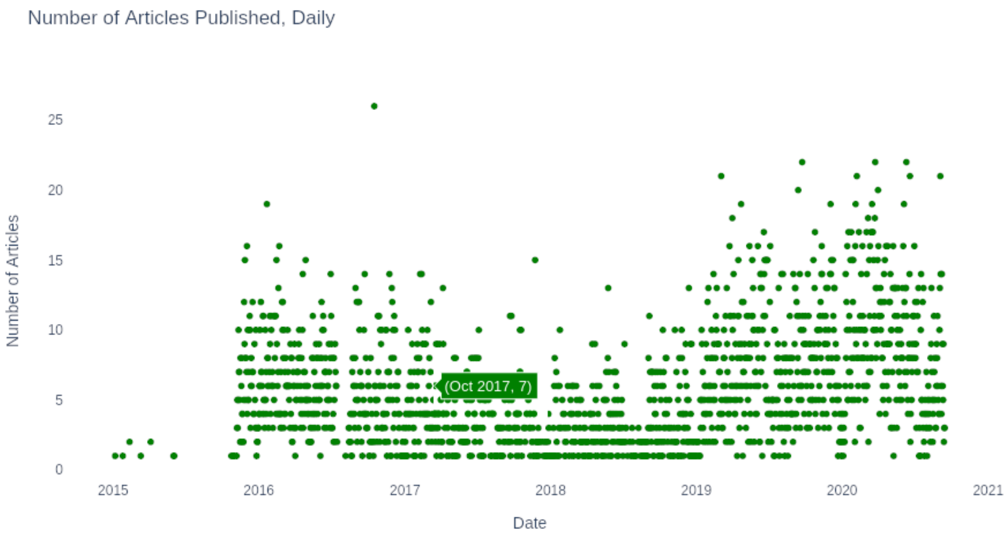


Figure 5.5: Duration of curated data in EUvsDisInfo dataset

This dataset is considered the gold dataset for fine-tuning the **Justification Generator** model using the text generation approach in the **Veracity Explanation** component of our framework. Even though the hackathon was about COVID-19 related disinformation, we choose to use this dataset as it contained more general claims that were established to be disinformation by a journalist annotator (see Section 6.5).

5.4 EUvsDisInfo Dataset Analysis

The analysis conducted on the EUvsDisInfo dataset was on the length of tokens in justifications to determine the average token length parameter for fine-tuning the text generation model. Additionally, this average length is a good indicator of the average length of justifications a human annotator will manually write. This token length is also used while generating the justifications, later.

[2] https://euvsdisinfo.eu/

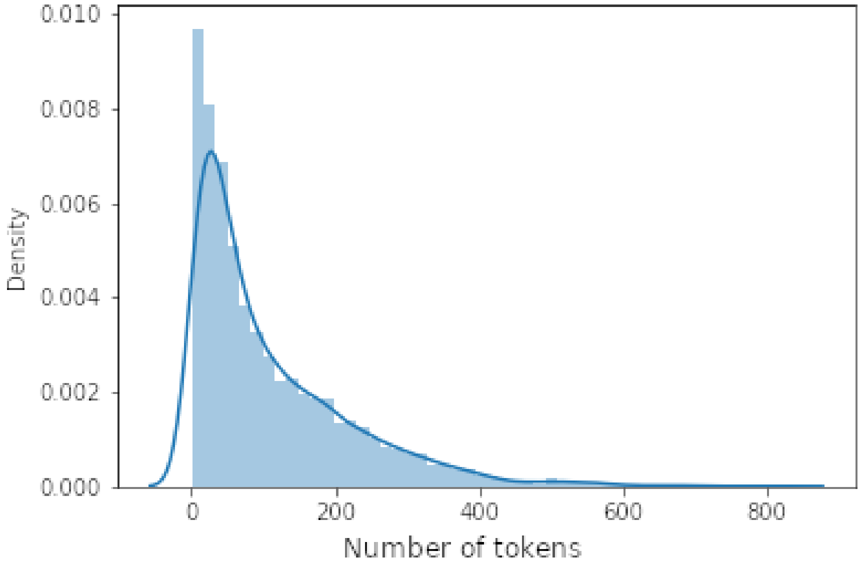


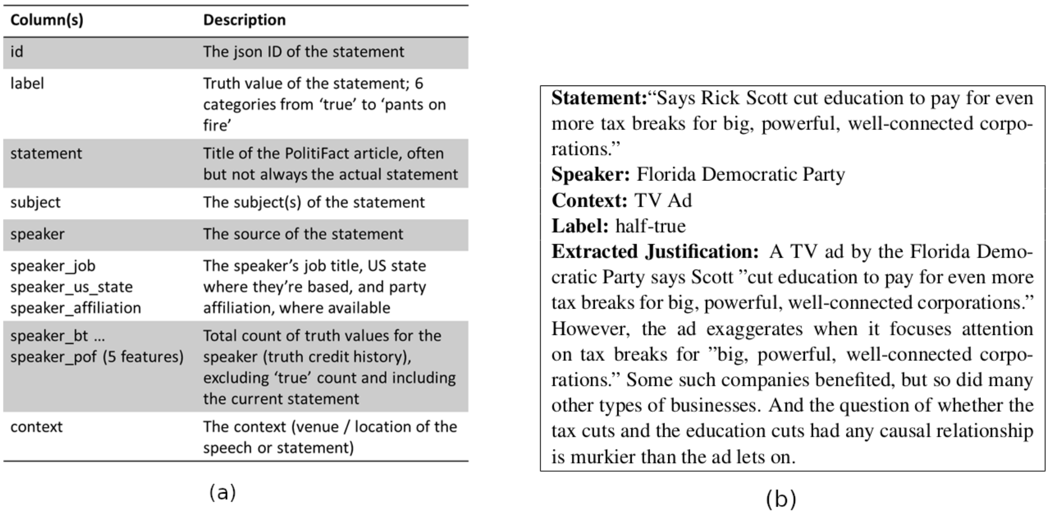
Figure 5.6: Number of words in manually written claim justification in EUvsDisInfo dataset

5.5 LIAR-PLUS Dataset

Wang (2017) introduced the LIAR dataset as a benchmark dataset for fake news detection and was an order of magnitude larger than the then-available datasets. In 2018, Alhindi et al. (2018) introduced the LIAR-PLUS dataset as an extension of the LIAR dataset, with an objective to empirically show that supplementing claims with additional meta-data including justification provides significant improvement in the classification task. The process involved enhancing the LIAR dataset by adding an automatically extracted full-text verdict report from PolitiFact. The features available in this dataset and an excerpt is shown in Figure 5.7. A detailed explanation of LIAR and LIAR-PLUS is in Section 2.1.

5.6 LIAR-PLUS Dataset Analysis

We conducted a few analysis studies on the dataset to understand its properties. The labels in the dataset are the same as LIAR. The truthfulness is distributed into six labels: pants-fire, false, mostly-false, half-true, mostly-true, and true. The dataset is relatively balanced between these labels, with the exception of 1,050 pants-fire cases and the rest falls in the range 2,063 and 2,638 samples. Using the default random forest classifier from Python library scikit-learn, a feature study is conducted and it is observed that the most important feature is the ‘subject’. As evident from Figure 5.8, the dark pink line indicates the subject feature and the light pink represent features from other columns. This study could indicate that



Column(s)	Description
id	The json ID of the statement
label	Truth value of the statement; 6 categories from 'true' to 'pants on fire'
statement	Title of the PolitiFact article, often but not always the actual statement
subject	The subject(s) of the statement
speaker	The source of the statement
speaker_job speaker_us_state speaker_affiliation	The speaker's job title, US state where they're based, and party affiliation, where available
speaker_bt ... speaker_pof (5 features)	Total count of truth values for the speaker (truth credit history), excluding 'true' count and including the current statement
context	The context (venue / location of the speech or statement)

(a)

Statement:"Says Rick Scott cut education to pay for even more tax breaks for big, powerful, well-connected corporations."
Speaker: Florida Democratic Party
Context: TV Ad
Label: half-true
Extracted Justification: A TV ad by the Florida Democratic Party says Scott "cut education to pay for even more tax breaks for big, powerful, well-connected corporations." However, the ad exaggerates when it focuses attention on tax breaks for "big, powerful, well-connected corporations." Some such companies benefited, but so did many other types of businesses. And the question of whether the tax cuts and the education cuts had any causal relationship is murkier than the ad lets on.

(b)

Figure 5.7: (a) LIAR-PLUS dataset features and (b) Excerpt from LIAR-PLUS dataset

during training the model learns the 'subject' feature prominently.

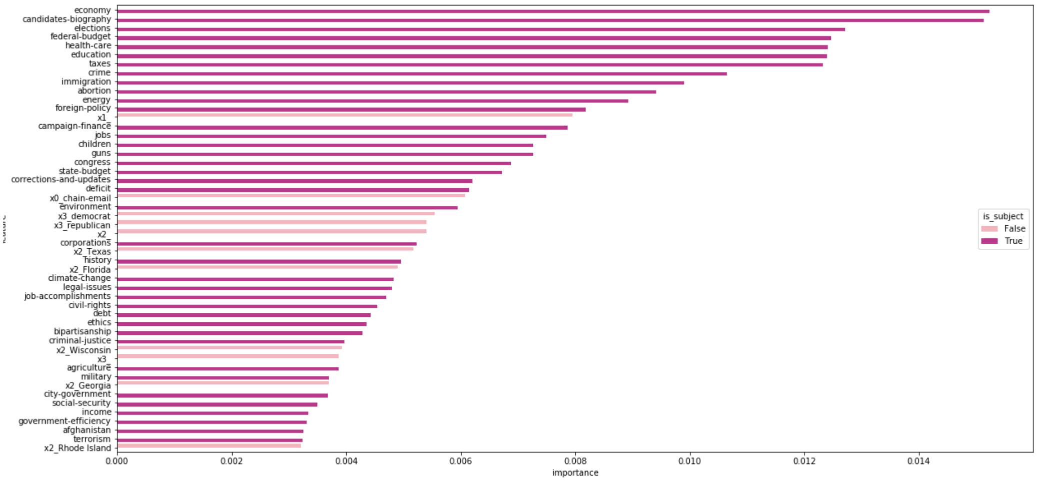


Figure 5.8: Feature study on LIAR-PLUS dataset

The veracity classification task in our framework is modeled as a binary classification task. Hence, we re-categorize the labels of this dataset to True (label 0) if the claims belonged to 'half-true', 'mostly-true', and 'true' and, to Fake (label 1) if they belonged to 'pants-fire', 'false', 'barely-true'. The distribution of samples still fairly remains balanced with 44.3% Fake and 55.7% True.

The LIAR-PLUS dataset is used for two purposes in our framework. First, the performance of the **Veracity Classifier** is compared by fine-tuning the model on various task-specific domain datasets. LIAR-PLUS is one of the datasets used for training the model along with EUvsDisInfo and TrueFake datasets. Second, we use the justifications present in the LIAR-PLUS dataset to fine-tune the summarization model in the **Justification Generator** module of the framework.

Chapter 6

Fake News Detection Framework

This chapter explains in detail the **Fake News Detection Framework** that is designed and implemented as a part of the thesis work. The framework introduced here is an end-to-end pipeline, where an input claim given by the user is fed to a fine-tuned classifier along with evidence, which predicts the veracity of the claim. Furthermore, the framework complements the classification output with additional information corresponding to each step in the pipeline along with justifications for the fake news detection process.

The high-level architecture of the **Fake News Detection Framework** is shown in Figure 6.1. It consists of three major components:

1. **Veracity Prediction:** Predicts the veracity of the input claim based on evidence.

2. **Veracity Explanation:** Generates the required explanations and justifications for the end-user, to better understand the fake news detection process.

3. **Journalists-in-the-Loop:** Allows end-users/journalists to give various feedback to the system thereby incorporating expert knowledge into the framework.

The three components are the fundamental pillars of this framework. Additionally, each component is appended with additional modules that allow achieving the desired functionality. The first few sections of this chapter explain these modules. The later sections explain the above-mentioned three pillars in detail.

6.1 Keyphrase Extractor

An input claim consists of important information that will help in retrieving evidence as well as predicting the veracity of the claim. When the framework receives an input claim, the first step is to collect the required information by extracting the relevant keywords from this claim. This is achieved by the **Keyphrase Extractor** module. It is built using the *'distilbert-base-nli-mean-tokens'* variant of

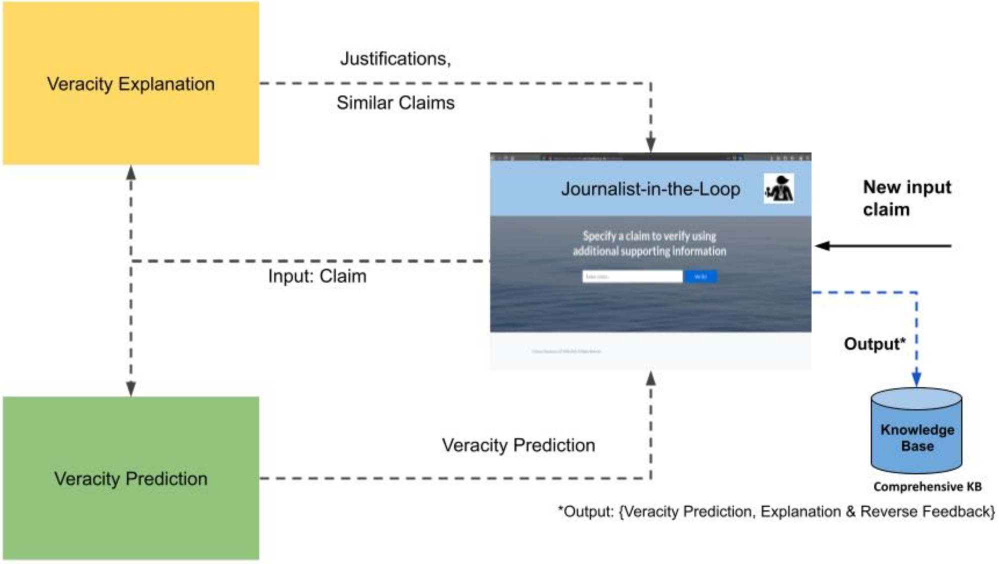


Figure 6.1: High-level architecture of Fake News Detection Framework

the DistilBERT model for natural language inference that belongs to the sentence-transformers. This model is known as KeyBERT (Grootendorst, 2020). It uses BERT-embeddings and cosine similarity to find sub-phrases in the input that are most similar to the claim itself. This modified version of BERT uses Siamese and triplet networks that derive semantically meaningful sentence embeddings to extract keywords (Reimers and Gurevych, 2019). The keyphrase extraction process involves extracting word embeddings for n-gram/phrases. Along with the extracted phrases, the model also generates the cosine similarity score, which is useful for understanding the importance of the keyphrases in reference to the input claim. This model allows extracting n-gram phrases from the input claim and hence optimizes the search query in the next component, which is the **Keyphrase Crawler**.

Let us consider an example input claim. This claim will be used as an example while explaining each module/component in the framework.

Input Claim: "PM launches key projects in poll-bound Assam, next stop Bengal"

With the ability of KeyBERT to extract n-gram phrases, we extract bigrams and trigrams for an efficient search of related evidences[1]. The following are the extracted keyphrases:

- **Bigrams phrases:** 'bound assam', 'pm launches', 'assam stop', 'stop bengal'
- **Trigrams phrases:** 'bound assam stop', 'poll bound assam', 'assam stop bengal', 'pm launches key'

[1] https://github.com/MaartenGr/KeyBERT

In our experiments with bigrams and trigrams search, we observed better search results obtained when we use trigrams. Hence, we use the extracted trigrams as keyphrases. Furthermore, this distilled architecture of BERT with minimal computational requirement and reduced number of layers ensure that KeyBERT does not introduce any performance overhead to the framework (40% smaller 60% faster (Sanh et al., 2019)).

6.2 Keyphrase Crawler

Our approach is based on verifying the claim using evidence curated from online sources. For extracting evidence we use a web crawler module, as it plays an essential role in efficiently collecting a corpus of web pages that are already indexed by search engines. We employ a similar approach followed while creating the **CompKB**, with a difference that here the articles searched for are based on keyword sub-phrases. Our web crawler uses trigrams as search criteria for retrieving online documents. This module is built using the Python *NewsAPI*[2] library. It is a JSON-based REST API used for searching news articles in 75,000+ news sources and blogs in the last 3 years. It allows refining the search criteria using keywords, sources, range of date, and sort by relevance options. The API key for using this library is obtained by registering with the organization. Even though there is a restriction of 100 search queries for 24 hours, we observe that by using the extracted trigram keyphrases from the input claim, sufficient requests are possible to retrieve the required amount of evidence for veracity prediction. We use the Python-based *Newspaper3K*[3] library to download the articles and their meta-data. The URLs scraped by the *NewsAPI* library are used to instantiate an *Article* class object. This object downloads the HTML content from the URLs and parses to extract information from these sources.

We define a few parameters to further refine and restrict the search results:

- **Article Limit per Keyword:** This parameter restricts the number of articles to be scraped per input keyphrase.
- **Total Number of Keywords:** This defines the total number of keyphrases (trigrams) generated by the **Keyphrase Extractor**.
- **Total Downloaded Article:** Multiplying the above two parameters defines the total number of articles downloaded.

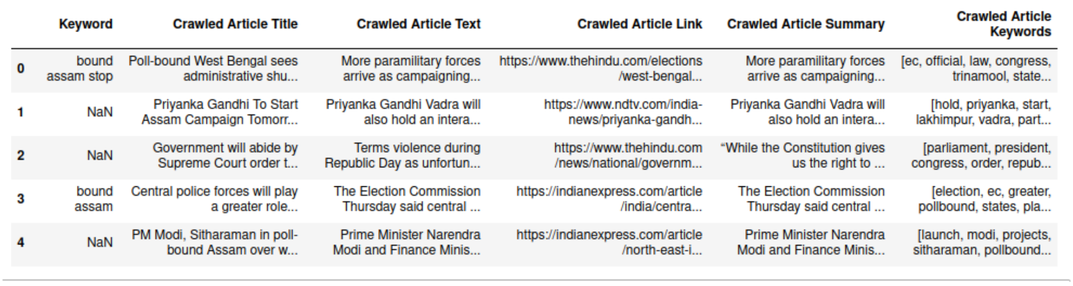
By defining these two parameters, referred as *article_limit_per_keyword* and *len_ download_articles* in the implementation helps us to place an upper bound on the total number of articles scraped while collating the evidence document set. This also ensures that the saved content has an upper bound on the storage requirement. Even though the downloaded articles require negligible storage space,

[2] https://newsapi.org/

[3] https://newspaper.readthedocs.io/en/latest/

we have encountered scenarios where the input keyphrases cause the crawler to scrape thousands of articles and eventually download them. Such an accumulation is harmful to the storage of the system hosting this framework, especially when the process is repeated daily for a longer time.

Additionally, we also use the *'sort by relevance'* option from *NewsAPI* library to ensure that the retrieved articles are relevant to the input keyphrases. Figure 6.2 shows the curated articles, in accordance with the example keyphrases:



	Keyword	Crawled Article Title	Crawled Article Text	Crawled Article Link	Crawled Article Summary	Crawled Article Keywords
0	bound assam stop	Poll-bound West Bengal sees administrative shu...	More paramilitary forces arrive as campaigning...	https://www.thehindu.com/elections/west-bengal...	More paramilitary forces arrive as campaigning...	[ec, official, law, congress, trinamool, state...
1	NaN	Priyanka Gandhi To Start Assam Campaign Tomorr...	Priyanka Gandhi Vadra will also hold an intera...	https://www.ndtv.com/india-news/priyanka-gandh...	Priyanka Gandhi Vadra will also hold an intera...	[hold, priyanka, start, lakhimpur, vadra, part...
2	NaN	Government will abide by Supreme Court order t...	Terms violence during Republic Day as unfortun...	https://www.thehindu.com/news/national/governm...	"While the Constitution gives us the right to ...	[parliament, president, congress, order, repub...
3	bound assam	Central police forces will play a greater role...	The Election Commission Thursday said central ...	https://indianexpress.com/article/india/centra...	The Election Commission Thursday said central ...	[election, ec, greater, pollbound, states, pla...
4	NaN	PM Modi, Sitharaman in poll-bound Assam over w...	Prime Minister Narendra Modi and Finance Minis...	https://indianexpress.com/article/north-east-i...	Prime Minister Narendra Modi and Finance Minis...	[launch, modi, projects, sitharaman, pollbound...

Figure 6.2: Crawled articles along with its additional extracted information

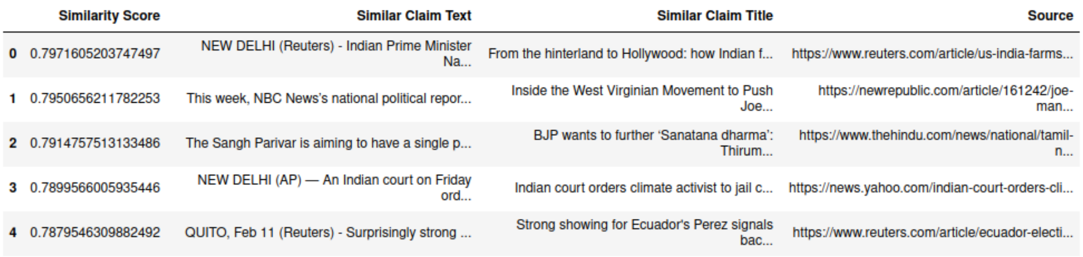
While curating the evidence set using this crawler, we collect the date of publishing, title, and textual content of the relevant news articles. Additionally, certain news websites also publish the summary of the article and the relevant keywords, which are added to our evidence set, if available, from the URL.

6.3 Similar Claim Extractor

When collating the evidence documents as an input for the veracity prediction as well as for generating justifications in our framework, it is essential that these documents must be similar to the content of the input claim. For this, we introduce a **Similar Claim Extractor** module to retain only those articles that are similar to the input claim. A similarity check is performed between the input claim and the retrieved document from online sources. The documents are then ordered from most similar to the least similar. We use the top-K strategy to extract the most K relevant documents.

This module is built using *SpaCy*, a Python-based natural language processing library. We use the largest English model (788 MB) *'en_core_web_lg'*[4]. *SpaCy* determines similarity by comparing word embeddings or word vectors, which are multi-dimensional representations of words. The similarity values range between 0 and 1 with 1 indicating that the inputs are the same. A snippet of the extracted and sorted evidence for the input user claim along with their similarity scores is shown in Figure 6.3. Detailed examples of similarity score and evidence are available in Appendix A.

[4] https://spacy.io/models/en



	Similarity Score	Similar Claim Text	Similar Claim Title	Source
0	0.7971605203747497	NEW DELHI (Reuters) - Indian Prime Minister Na...	From the hinterland to Hollywood: how Indian f...	https://www.reuters.com/article/us-india-farms...
1	0.7950656211782253	This week, NBC News's national political repor...	Inside the West Virginian Movement to Push Joe...	https://newrepublic.com/article/161242/joe-man...
2	0.7914757513133486	The Sangh Parivar is aiming to have a single p...	BJP wants to further 'Sanatana dharma': Thirum...	https://www.thehindu.com/news/national/tamil-n...
3	0.7899566005935446	NEW DELHI (AP) — An Indian court on Friday ord...	Indian court orders climate activist to jail c...	https://news.yahoo.com/indian-court-orders-cli...
4	0.7879546309882492	QUITO, Feb 11 (Reuters) - Surprisingly strong ...	Strong showing for Ecuador's Perez signals bac...	https://www.reuters.com/article/ecuador-electi...

Figure 6.3: Extracted similar evidence from online sources

6.4 Veracity Prediction

The **Fake News Detection Framework** consists of three main pillars where **Veracity Prediction** is the first one. This component embeds one of the core objectives of the framework: to determine the veracity of a claim. Figure 6.4 illustrates the detailed architecture of this component. As observed from the architecture, this component uses **Keyphrase Extractor** to extract keyphrases as trigrams from the input claim and passes it to the **Keyphrase Crawler** to search the online news sources to retrieve web pages as documents. These documents are processed by the **Similar Claim Extractor** to extract the top-K similar documents and are composed into an evidence set. This evidence set along with the input claim is presented to the **Veracity Classifier** for predicting the veracity of the input claim. The core component of **Veracity Prediction** is the **Veracity Classifier**, with its purpose to determine the truthfulness of the input claim.

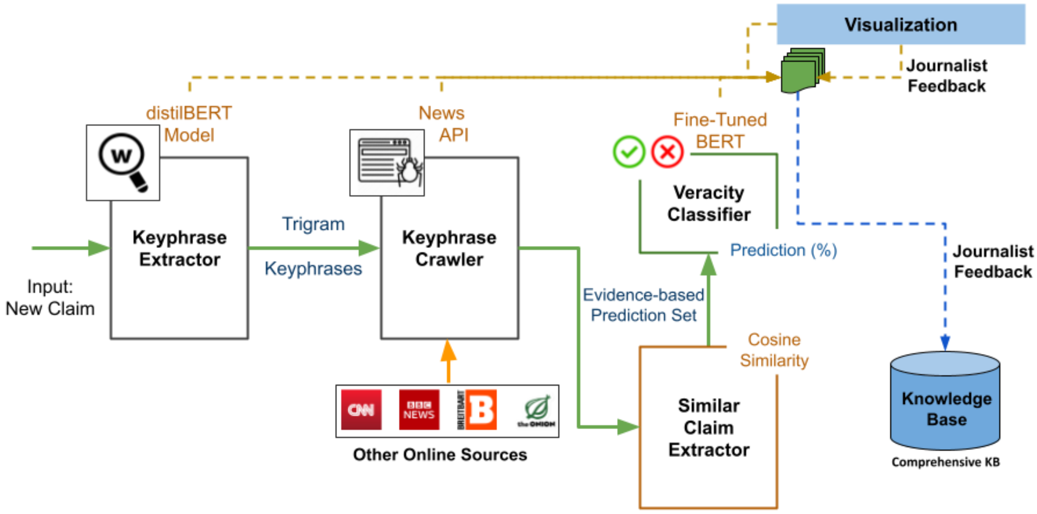


Figure 6.4: Low-Level architecture of Veracity Prediction

Veracity Classifier

Fact-verification methods using deep learning models are well-explored in the natural language processing domain and there is much attention to transformer-based models. As explained in Section 3.3, we use the BERT transformer model to automate the veracity prediction as well as leverage the ability of transfer learning from pre-trained models. The following subsections explain in detail the task for which BERT is fine-tuned, the fine-tuning process, and the integration of the fine-tuned model within the framework.

Task Definition

The first step in fine-tuning a pre-trained model is to define the task for which it is required to be fine-tuned. We define the task for the **Veracity Classifier** model as a classification problem.

Task Definition: Under the assumption that the input claim is true, the model requires to predict the truthfulness of the input claim based on provided evidence.

We fine-tune BERT using the TrueFake Dataset (see Section 5.1), since it comprises a collection of true and fake labeled claims and its justifications, with a training/validate/test split of 60%, 20%, 20% respectively. For further understanding of the performance of the model, we fine-tune BERT on additional datasets as explained in detail in Section 7.1.

Fine-Tuning BERT

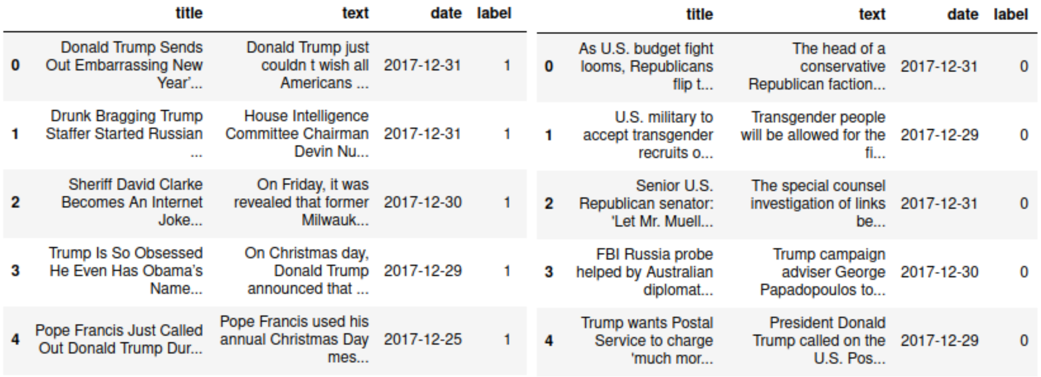
BERT allows fine-tuning to any downstream task with much ease by swapping out the appropriate inputs and outputs. BERT, with its self-attention mechanism, effectively includes bidirectional cross attention between the input sentences, thus making fine-tuning less computationally demanding than training from scratch. The prerequisite for using the BERT model is the *transformers*[5] package from Hugging Face, which is a PyTorch-based interface for working with any transformer model. The *'uncased-based'* version of the BERT model is our baseline model.

For reference, a snippet of the training dataset is shown in Figure 6.5. It consists of three columns: '*title*' - the claim, '*text*' - the justification of the claim, and '*label*' - the label stating whether the claim is fake (label 1) and true (label 0). To use the pre-trained BERT model, the input requires to be tokenized for two reasons:

1. The model has a specific fixed vocabulary.
2. It has a special way of handling out-of-vocabulary words.

We use the *'bert-base-uncased'* as the tokenizer, which is based on *WordPiece*. The tokenizer takes the input sequences and generates the respective token IDs.

[5] https://github.com/huggingface/transformers



	title	text	date	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	2017-12-31	1
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	2017-12-31	1
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	2017-12-30	1
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	2017-12-29	1
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	2017-12-25	1

	title	text	date	label
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction...	2017-12-31	0
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fi...	2017-12-29	0
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links be...	2017-12-31	0
3	FBI Russia probe helped by Australian diplomat...	Trump campaign adviser George Papadopoulos to...	2017-12-30	0
4	Trump wants Postal Service to charge 'much mor...	President Donald Trump called on the U.S. Pos...	2017-12-29	0

Figure 6.5: Snippet of fake (label 1) and true (label 0) articles in TrueFake dataset

These IDs are integer sequences of the inputs and attention mask, consisting of ones and zeros, where the zeros correspond to the masked tokens. The BERT tokenizer has a maximum token length limit of 512 tokens. Adapting this parameter based on the input dataset ensures that all the sentences are tokenized to the same length. To determine the maximum length while tokenizing the '*title*' and the '*text*' inputs (see Appendix B, Figure 6.5), the distribution of the lengths of the sequences are plotted as histograms. Values of 100 and 400 are chosen as the maximum length for '*title*' and '*text*' inputs, respectively since the majority of the inputs were around these respective lengths. As a final step, before training, these integer sequences are converted to tensors, allowing the encoding of multi-dimensional representations of input tokens IDs.

BERT Architecture

The pre-trained *'bert-base-uncased'* model encompasses 12 transformer layers, 768-hidden, 12-heads, a total of 110M parameters trained on lower-cased English text. The weights of the pre-trained encoder are frozen and only the weights of the head layers are optimized. This is done by setting the *requires_grad* attribute to *false* in the encoder parameters. The model architecture is defined as follows:

- Dropout Layer: 20%
- Activation Function: ReLU
- Dense Layer 1: Linear (768, 512)
- Output Dense Layer: Linear (512, 2)
- Output Activation Function: Softmax

The AdamW optimizer (Loshchilov and Hutter, 2017), which implements gradient bias correction as well as weight decay, is defined with a learning rate of 1e-5.

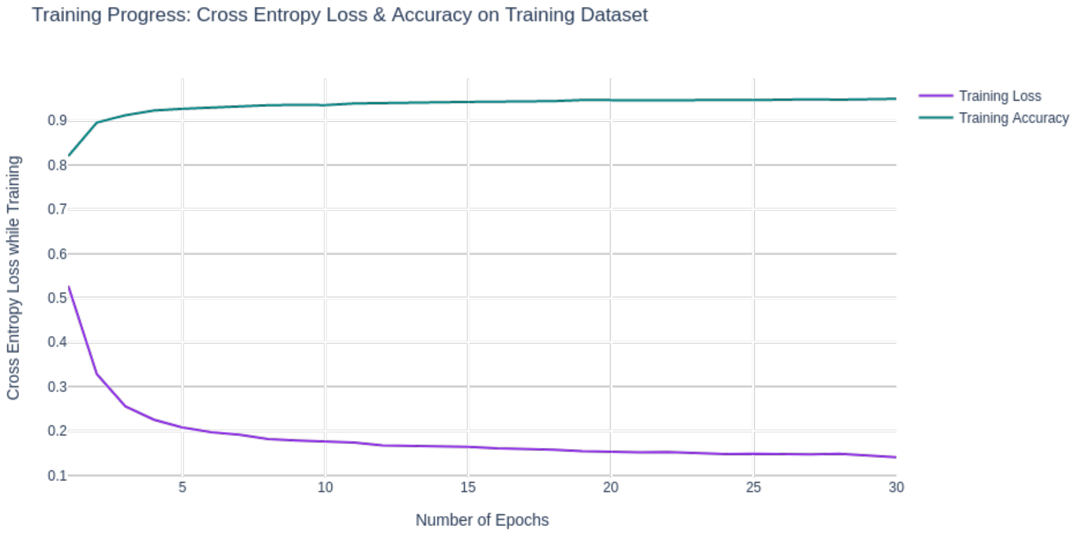


Figure 6.6: Fine-Tuning: Cross entropy and accuracy on TrueFake training dataset for 30 epochs using the pre-trained BERT model

Training: The model hence defined is trained on 60% of TrueFake Dataset as training data, with a train/validation split of 80/20 on a batch size of 16 for 30 epochs. At the end of each epoch, the model evaluates using the validation data and minimizes a cross-entropy loss function. Figures 6.6 and 6.7 show the accuracy and cross-entropy loss on the training and validation dataset. Post-training, the model is saved for further use in the framework. Additionally, various experiments using different datasets are conducted to understand the performance of the model. Refer Section 7.1 for detailed explanation about the same.

Testing: By reloading the learned weights of the fine-tuned model, the predictions are made on the test set. The following is the score metrics on the test set:

- Loss: 0.04049
- Accuracy: 0.9569
- F1-Score: 0.9576

Framework: This fine-tuned model is pre-loaded within the framework. The input claim along with the evidence set collated using the **Keyphrase Crawler** and **Similar Claim Extractor** is fed to this fine-tuned model. The model outputs the prediction accuracy and is interpreted as follows: As an example, when the model generates 0.47 as the prediction accuracy, this means assuming that the input claim is true, the model predicts it to be true by 47%. We decided to use the percentage-based prediction accuracy as this enables a subjective veracity

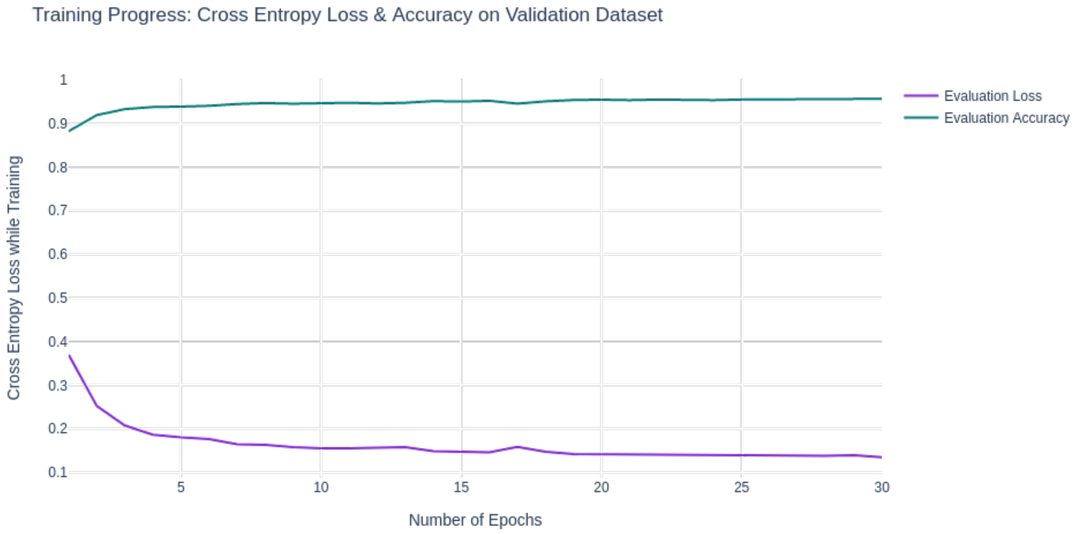


Figure 6.7: Fine-Tuning: Cross entropy and accuracy on TrueFake validation dataset for 30 epochs using the pre-trained BERT model

classification of the input claim. We also observed that from various online manual fact-checking systems, the explanations that justify the veracity of input claims are subjective in nature. The objective of our framework is to provide end-user with assistive information and not a final verdict of the veracity, and hence we use this percentage-based prediction accuracy approach for further end-user (domain-expert) inferences.

6.5 Veracity Explanation

For our **Fake News Detection Framework** to successfully accomplish the fake news detection process, it is not only important to predict the veracity of the claims but also is required to be self-sufficient by generating explanations for the predicted veracity. Including explanations in the framework renders useful to the end-users/journalists by providing them with indicative evidence for the veracity of the claim. As suggested by Graves (2017), journalists work with a set of guidelines, and generating justifications as explanations is one of the key aspects of their guidelines. Hence, the second important pillar of our framework is **Veracity Explanation**, which has an overall objective to generate explanations for the veracity of the claim. To accomplish this objective we focus on two main directions:

1. Generate explanations similar to human-annotated justifications in a fact-verification system. We refer to these explanations as *justifications* in our framework.

2. Collect relevant *evidence* from online sources as indicative resources to the

generated justification.

Figure 6.8 illustrates the detailed architecture of the **Veracity Explanation** component. The **Justification Generator** in the architecture is built to accomplish the objective of generating justifications similar to human annotators. It is common for fact-checkers/journalists, referred to as human annotators, to manually write justification about the veracity of the claim. Examples of such justifications can be found in fact-checking websites like PolitiFact, also with evidence supporting their justifications. The rest of the modules in the architecture curate the relevant online evidence.

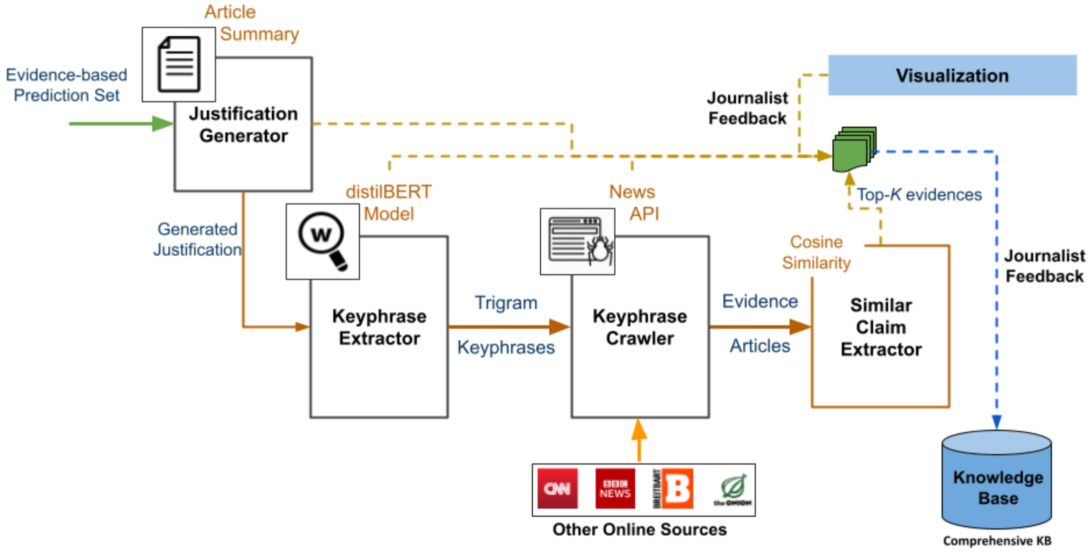


Figure 6.8: Low-level architecture of Veracity Explanation

We adapted two well-studied approaches for generating explanations: text summarization and text generation. By formulating explanation generation as 1) text summarization task and 2) text generation task, we follow the transfer learning approach by fine-tuning pre-trained deep learning models for the respective tasks. The following subsections explain these two tasks in detail.

Justification Generator

The **Justification Generator** component aims at automatically producing explanations to complement the most elaborate journalistic process in fact-verification: writing justifications for the veracity prediction. We follow two approaches to generate justifications using deep learning models:

- **Approach 1:** Text Summarization
- **Approach 2:** Text Generation

Both approaches have their advantages and disadvantages. The following subsections explain both approaches in detail.

Approach 1: Text Summarization

A well-studied task in Natural Language Processing (NLP) is text summarization and they have been used in different domains like news article summarization and scientific paper summarization. Text summarizations intent to capture the key information present in a long text based on extractive or abstractive methods. Extractive methods that involve cropping out and stitching together portions of text to produce condensed versions, are arguably well suited for extracting the most relevant information in a long text but may lack fluency and coherency compared to human-generated summaries. On the other hand, abstractive methods involve paraphrasing contents of the original text and may have the chance to not capture the important information. In this thesis, we focus on the abstractive summarization approach. We employ the state-of-the-art transformer model T5 that uses an abstractive summarization algorithm. The T5 transformer defines the NLP tasks in a unified text-to-text framework making it apt for using it as a summarization model in our framework (Raffel et al., 2019). The model is adapted to our task requirement by fine-tuning it using a task-specific dataset. The automatic summarization in our framework is achieved by feeding the article text to the fine-tuned model and producing summaries by auto-regressive decoding. Refer to Section 3.4 for a detailed explanation of the T5 architecture.

Task Definition

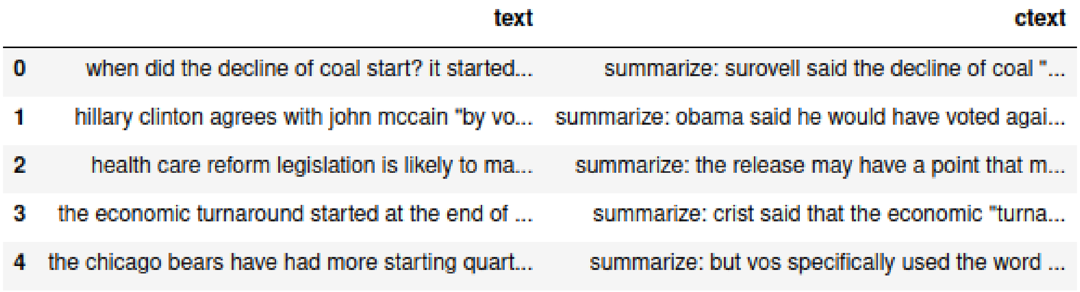
We begin by defining the task for which the T5 model is fine-tuned using a dataset that consists of human-annotated justifications.

Task Definition: The text summarization model requires the creation of an abstractive summary from a set of evidence articles provided as input.

To achieve this task, we fine-tune the T5 model using the LIAR-PLUS dataset. This dataset contains human-annotated justifications for its respective claims, allowing the model to learn representations that will enable it to produce abstractive summaries.

Fine-Tuning T5

The text-to-text framework in which the T5 model is pre-trained allows it to be fine-tuned to any downstream task that can be formulated as the input sequence being transformed to an output sequence. Our summarization task involves transforming the input articles, which are long sequences of text, into abstractive summary, short, compressed version of the input text. We use the training and validation set of the LIAR-PLUS dataset as the fine-tuning dataset with a train and validation split of 80% and 20% respectively. For reference, Figure 6.9 illustrates a snippet of the training data.



	text	ctext
0	when did the decline of coal start? it started...	summarize: surovell said the decline of coal "...
1	hillary clinton agrees with john mccain "by vo...	summarize: obama said he would have voted agai...
2	health care reform legislation is likely to ma...	summarize: the release may have a point that m...
3	the economic turnaround started at the end of ...	summarize: crist said that the economic "turna...
4	the chicago bears have had more starting quart...	summarize: but vos specifically used the word ...

Figure 6.9: Snippet of LIAR-PLUS dataset prepared for summarization task

The prerequisite for using the T5 model is the *transformers* package from Hugging Face, which is a PyTorch-based interface for working with any transformer model. We use the *T5 tokenizer*, which constructs XLNET tokenizer based on SentencePiece, an unsupervised text tokenizer and detokenizer[6]. It tokenizes the data in 'text' and 'ctext' columns. Columns 'text' and 'ctext' contain headlines and the complete text from the articles, respectively. While pre-processing the article text to be summarized, the keyword 'summarize' is appended to the beginning of every article text. This is a format required for the T5 summarization dataset. The outputs produced by the tokenizer, IDs, and masks of actual text and target summary text, are passed to the model for fine-tuning, using the data loader as per the defined batch size. The labels for the language model are calculated from the target IDs. For each epoch, the loss value is determined and is used to optimize the weights of the network. The training dataset is further split into 80/20 so that we have 20% of data to be used for the validation run. During validation runs, the weights of the model are not updated. Finally, the generated text and the original summary are decoded from tokens to produce the respective texts.

T5 Architecture

The base model consists of raw hidden states without any specific attention head on top. We use the variant known as *T5forConditionalGeneration* which has a language model head on top, which adds a linear layer with weights tied to input embeddings. The language model head generates text based on the pre-training of the T5 model.

The following hyper-parameters were selected by taking into account the computation power and resources at hand:

- Training Batch Size = 2 (default: 64)
- Validation Batch Size = 2 (default: 1000)

[6] https://github.com/google/sentencepiece

- Training Epochs = 30 (default: 10)
- Validation Epochs = 4 (default: 10)
- Learning Rate = 1e-4 (default: 0.01)
- Seed = 42 (default: 42)

The AdamW optimizer (Loshchilov and Hutter, 2017), which implements gradient bias correction as well as weight decay, is defined with a learning rate of 1e-4.

Training: The number of epochs, tokenizer, model, device details, dataloader, and optimizer is passed to the train function to fine-tune the model. The model is trained using 80% of the LIAR-PLUS combined train and validation dataset. The batch size used here is 2 and we trained the model for 30 epoch. Any other hyper-parameter setting than the above-mentioned led to memory issues on the CUDA. Given the computational resources, the training for 30 epochs took more than 20 hours. This indicates that the state-of-the-art deep learning models are computationally intense. The plots in Figure 6.10 show the cross-entropy loss in the training dataset. Post-training, the model is saved for further use in our framework. We conducted qualitative and quantitative evaluation studies for understanding the performance of the model with and without fine-tuning (see Section 7.2)

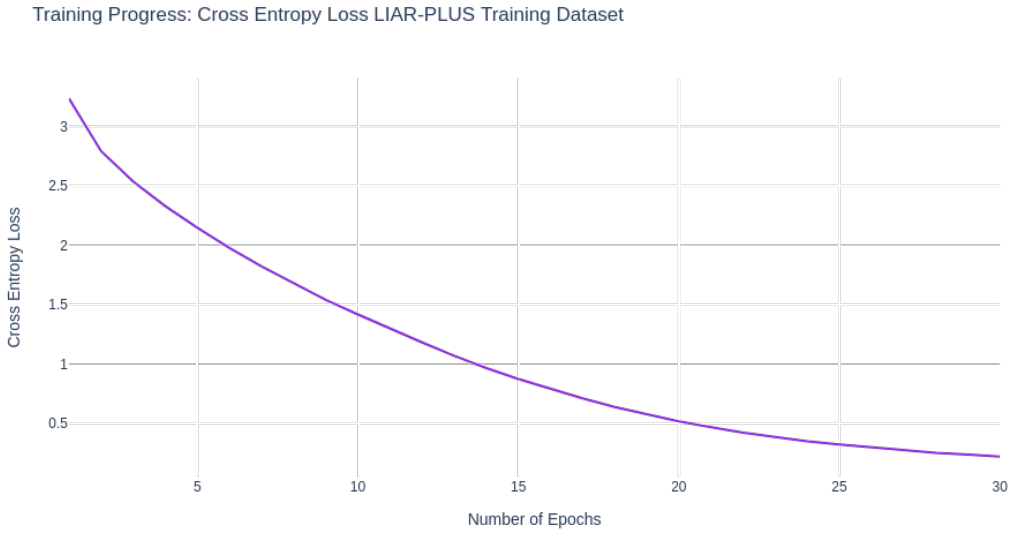


Figure 6.10: Fine-Tuning: Cross entropy loss on LIAR-PLUS training dataset for 30 epochs using the pre-trained T5 model

Testing: By reloading the learned weights of the fine-tuned T5 model, we generate the summaries using the test set. The generated summaries on the test set are evaluated using the ROUGE score, which is presented in Section 7.2.

Framework: An important requirement while generating summaries is that we provide the model with the correct input. The evidence set curated from online sources based on the top-K similarity method is provided as the input to the T5 fine-tuned model. Thus, the generated summaries by the model are relevant to the claim and can be used as indicative justification for the end-user/domain expert to create their final justification.

Approach 2: Text Generation

Deep learning techniques are used to learn general representations of textual content and are employed in various text generation tasks such as poetry, movie scripts, music composition, and so on. The journalistic process of fact-verification involves manually writing justifications for the veracity of a claim. As a second approach, we formulated this process to be a text generation task. We aim to cover two objectives with this task:

1. To automate the process of manual justification writing and generate justifications that will help end-users/journalists to effectively articulate their final justification.

2. Evaluate the ability of pre-trained deep learning model to learn the style of human-annotation reference for text generation and generate justifications that are closer in the domain, by model fine-tuning.

We experimented using the state-of-the-art Generative Pre-Trained Transformer-2 (GPT-2) model for text generation from OpenAI[7]. The following are the reasons for which we chose this model:

- GPT-2 is pre-trained on 8 million web pages that approximates to 40GB of text. Since our framework uses the content from news organization websites as sources and hence, we can benefit from the model learnings as it is also pre-trained on web page texts.

- When fine-tuning, GPT-2 allows generating abstractive text using a relatively small dataset, allowing us to leverage transfer learning.

Task Definition

The first step is to define the task for which the model is to be fine-tuned. We define this task as a language generation task, where sequences of English sentences are to be generated, using the input claim from the end-user as the seed.

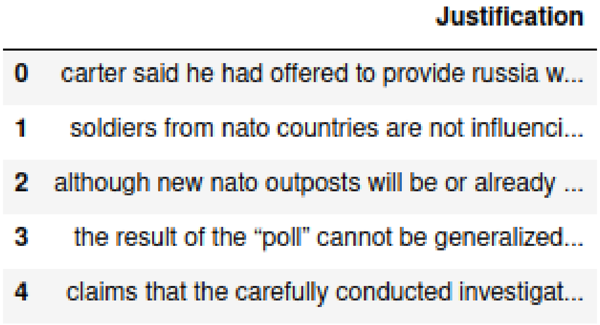
[7] https://github.com/openai/gpt-2

Task Definition: Given the input claim as a seed, the model is required to generate abstractive explanations in a style similar to the reference human annotations.

These generated sentence sequences are called the justification to the input claim. The objective of the task here is for the model to generate veracity justifications similar to human justifications. The justifications from the EUvsDisInfo dataset is used for this task (see Section 5.3), with a training/validate/test split of 60%, 20%, 20% respectively.

Fine-Tuning GPT-2

For us to evaluate the model in our framework, it is important to have the required dataset for fine-tuning this model. The EUvsDisInfo dataset serves this purpose as it contains manually written justification for a collection of claims curated by the author. It is to be noted that this dataset consists of only 9551 entries, which is relatively small compared to the size of the dataset used for pre-training GPT-2. Collecting a human-annotated dataset is an expensive process and we benefit from this dataset, as this was available to us as part of the CLARIN hackathon (see Section 5.3). Apart from the text generation task, we also aim to understand the fine-tuning process of such a massive pre-trained model. Figure 6.11 shows a snippet of the pre-processed dataset, which contains only the justifications from the original dataset.



	Justification
0	carter said he had offered to provide russia w...
1	soldiers from nato countries are not influenci...
2	although new nato outposts will be or already ...
3	the result of the “poll” cannot be generalized...
4	claims that the carefully conducted investigat...

Figure 6.11: Snippet of justification from EUvsDisInfo dataset

The fine-tuning process is similar to the process followed for the BERT model. The data is pre-processed and tokenized using the *GPT-2 tokenizer*. This tokenizer is based on byte-level byte-pair-encoding and is trained to treat spaces like parts of a token. Hence, the tokenizer requires special tokens to be added to indicate parts of the input sequences. The beginning and end of sequence is indicated with $< |startoftext| >$ and $< |endoftext| >$ tokens respectively and $< |pad| >$ for padding. These tokens are also assigned special token IDs. Padding is required to ensure that all the input sequences are of the same length. The data is split into training and validation sets, loaded using DataLoaders.

GPT-2 Architecture

For fine-tuning, the default configuration from the pre-trained *GPT2LMHeadModel* is used, which has a language model head on top of the bare GPT-2 base model. This head is a linear layer with weights tied to the input embeddings. The implementation that we use is based on PyTorch.

The following hyper-parameters were selected by taking into account the computation power and resources at hand:

- Training Batch Size = 2
- Training Epochs = 10
- Learning Rate = 5e-4 (default: 0.01)
- Seed = 42 (default: 42)

These parameters are loaded along with the model and pushed to GPU for training. The default GPT-2 model architecture is used here. The AdamW optimizer with a learning rate of 5e-4 is defined.

Training: The model, along with its parameters are trained on 60/20 training and validation dataset, respectively for 10 epochs with a batch size of 2. Every epoch evaluates the model and minimizes the cross-entropy loss. Any other hyper-parameter setting than the above-mentioned led to memory issues on the CUDA. Given the computational resources, the training for 10 epochs took approximately 9 hours. This indicates that state-of-the-art deep learning models are computationally expensive. The plots in Figure 6.12 show the cross-entropy loss for training and validation. Post-training the model is saved for further use in our framework. We conducted qualitative and quantitative evaluation studies for understanding the performance of the model with and without fine-tuning (see Section 7.2).

Testing: With the learned weights of the fine-tuned model, the sequence of words is generated for a length of 165, which is the average justification length in the input dataset. We observe that the vocabulary of the generated text is biased towards the content from the fine-tuning dataset. This indicates that the model learns features specific to the training dataset. We conducted a qualitative analysis to understand if the generated text imitates the input reference style. For quantitative evaluation, the generated summaries on the test set are evaluated using the ROUGE score. A detailed study can be found in Section 7.2.

Revisiting the main objectives of **Veracity Explanation**, the first objective to generate explanations is accomplished by experimenting with the above two approaches. In our quantitative and qualitative evaluation studies, we observe that the text summarization approach generates better justifications compared to the text generation approach. Hence, we use the fine-tuned T5 model and pre-load this model in the **Justification Generator** module of our framework (see Figure 6.8)

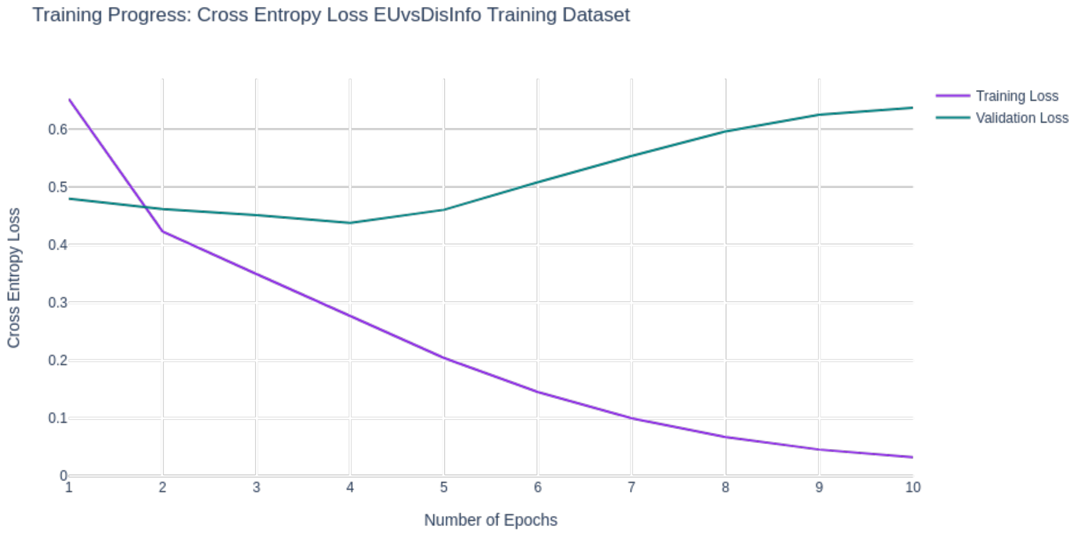


Figure 6.12: Fine-Tuning: Cross entropy loss on EUvsDisInfo training and validation dataset for 10 epochs using the pre-trained GPT-2 model

The second objective of this module is to collect the relevant evidence as indicative resources for the generated justification, which is achieved as follows:

The generated justification is sent to the **Keyphrase Extractor**, which extracts trigrams as keyphrases. These keyphrases are used by the **Keyphrase Crawler** to search the online sources and it retrieves a set of online documents, as explained in Section 6.2. The **Similar Claim Extractor** (see Section 6.3) selects the most similar documents to the input claim using the top-K strategy. This final set is presented to the end-user as the evidence set for the above-generated justification. It is important to note that, the keyphrases (trigrams) are also presented to the end-user for better understanding the evidence search.

We emphasize the importance of feedback from the end-users/domain experts/journalists and the following section explains the way we achieve the same.

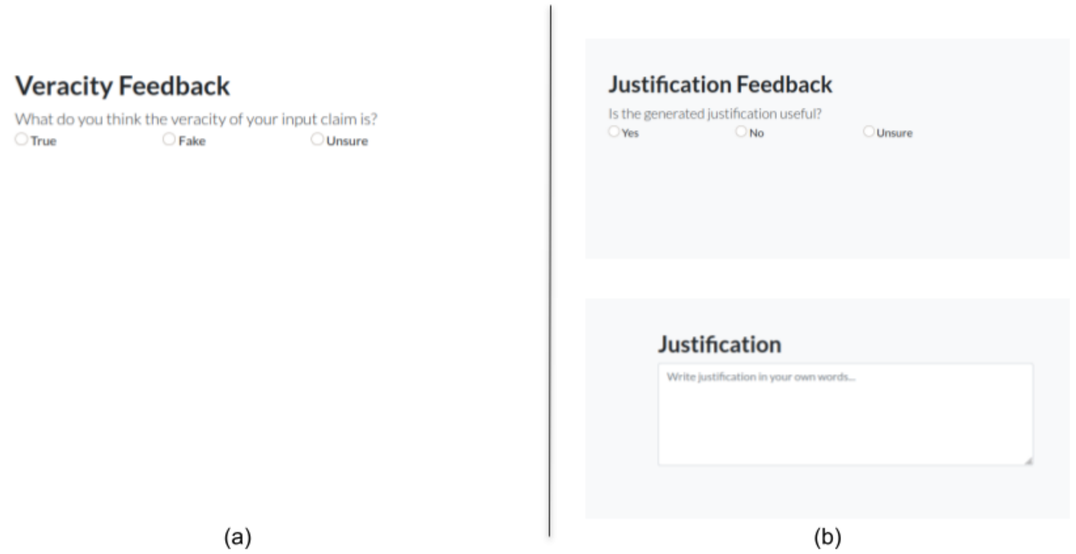
6.6 Journalist-in-the-Loop

We described earlier, in Section 2.4, that AI-enabled systems perform better when the expertise of end-users are embedded into these systems. We follow an approach where our framework supports the end-users, in our case journalists, to give feedback about the veracity prediction and explanation. This feedback forms the third pillar of our framework known as **Journalist-in-the-Loop**. We aim to achieve the following objectives with this approach:

- Create a journalist-centered **Fake News Detection Framework** to support them in researching and verifying news.

- Continuously enhance our **Comprehensive Knowledge Base** (CompKB, see Section 4.1) with domain expert knowledge from journalists.

We achieve the above two objectives by providing feedback options in the user interface of our framework. Figure 6.13 illustrates the feedback elements of the interface.



Veracity Feedback

What do you think the veracity of your input claim is?

○ True ○ Fake ○ Unsure

Justification Feedback

Is the generated justification useful?

○ Yes ○ No ○ Unsure

Justification

Write justification in your own words...

(a)

(b)

Figure 6.13: Feedback options in the user interface of Fake News Detection Framework (a) Veracity feedback from end-user irrespective of the prediction from the framework and (b) Feedback in utility of the explanation and optional textual justification feedback

Veracity Prediction: We ask the end-user about the veracity of the input claim that they checked for, as per their knowledge. This veracity (fake/true/unsure) is independent of the prediction that our framework has made. When domain experts such as journalists use our framework, they might be aware of the truthfulness of the news. This awareness is captured in our framework. Including this information has two benefits. We update our **CompKB** with this information and, this enhances the quality of the knowledge base. Further, we use this feedback in our periodic re-training of the **Veracity Classifier** model. This helps in enhancing the model performance as well as the quality of the knowledge base.

Veracity Explanation: Justifications are automatically generated by the fine-tuned model and are based on the evidence curated by the framework. We collect feedback regarding the usefulness of the generated justification. This feedback allows us to evaluate the model performance from a qualitative perspective and also gives us a fair understanding of the quality of curated evidence. As more elaborate feedback from the end-user, we also provide optional textual feedback, where they can manually write the justification for the claim in their own words, similar to the manual fact-checking process.

Comprehensively, this component allows feedbacks from end-users and helps us to create an end-user inclusive framework. There are various challenges introduced in such a framework, which is explained in Section 8.3.

Referring to the high-level architecture of our framework (Figure 6.1), the information flow is as follows: The input claim is collected from the end-user through the user interface and is passed to the **Keyphrase Extractor** which extracts trigram keyphrases. The **Keyphrase Crawler** then searches online sources for articles that contain these keyphrases and scrapes these articles. By placing an upper bound on the number of scraped articles and using a top-K strategy, the K most similar articles are curated as an evidence set by the **Similar Claim Extractor** module. This evidence set along with the input claim is fed into the **Veracity Classifier** for predicting the percentage of truthfulness. The **Veracity Explanation** component uses the **Justification Generator** to abstractively summarize the evidence documents and is passed to the **Keyphrase Extractor** for trigram keyphrases. The **Keyphrase Crawler** again retrieves the relevant news articles and collates the top-K similar articles using **Similar Claim Extractor**. The outputs from each component/module are then displayed to the end-user through the user interface. The end-user is required to give feedback on the veracity prediction and justification, with an optional manual text entry for justification. Once the user submits the feedback, our framework processes the feedback and persists the same. The persisted information consists of outputs from all the components/-modules as a collection in the format: 'Input Claim', 'Input Claim Keyphrases', 'Top-K Similar Evidence Set', 'Veracity Classifier Value', 'Justification', 'Justification Keyphrases', 'Top-K Similar Justification Evidence', 'Veracity Classifier Feedback', 'Justification Usability Feedback', 'Justification Textual Feedback'. We use the **Keyphrase Crawler** twice to search for as many articles depending on the input claim and generated justification, hence presenting the end-user with evidence that could be useful to make the final verdict and justification for their input claim.

The user interface and framework implementation are explained in Section 6.7. We conducted various evaluation studies and present the observations in the next chapter, along with a real news example to illustrate the framework in action (see Appendix D).

6.7 Implementation

This section explains the details of the software and hardware used for implementing the **Fake News Detection Framework**. The implementation is available as a GitHub repository[8].

[8] https://github.com/vksoniya/fakenewsdetectionframework.git

Hardware

The deep learning models are trained using GeForce RTX/RTX and Titan Xp GPU cores.

Software

The operating system is Ubuntu 18.04.5 LTS (bionic) and Jupyter Notebooks are used for Python implementation. The models are implemented using the PyTorch and Python3 environment.

The datasets for training, validation, and testing are available in comma-separated values (csv) and tab-separated values (tsv) formats. The fine-tuned models are saved using the torch.save() in the common PyTorch convention using '.pt' extension. The function saves the *state_dict* of the model, allowing flexibility for later restoration. The output from each claim check is persisted in csv format.

The web server is currently hosted on the GPU server in our Lab. A prerequisite when using the framework is a GPU device as the models used require memory and processing capacity larger than a CPU. For hosting this framework, a hosting server with public access and minimum requirements such as Ubuntu OS, 16GB RAM, GPU processors, and 4GB of disk space is needed. These are indicative requirements and depending on the hosting server capacity, the response of the server will differ.

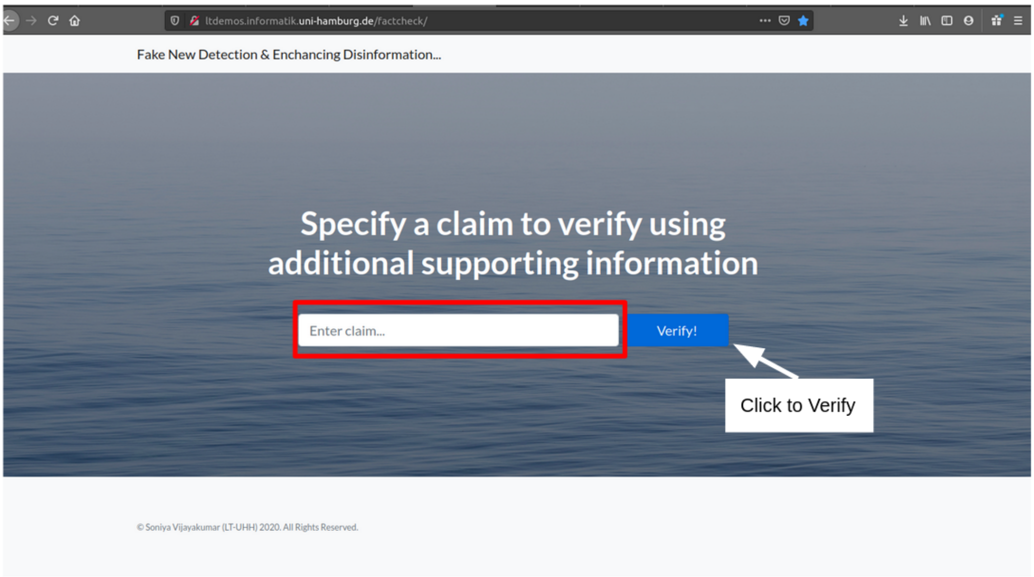
Visualization

The user interface for this framework is built in the **Visualization** module, which is implemented using *Flask*[9] based server rendered using HTML5, CSS, and JQueries. Flask is a micro web framework in *Python* that depends on *Jinja*[10] and *Werkzeug*[11] and delivers a lightweight WSGI web application framework. We decided to use Flask for its ease of use and built capacity without much overheads and dependencies like a full-stack front-end application. The screenshot of the interface is shown in Figure 6.14

[9] https://palletsprojects.com/p/flask/

[10] https://palletsprojects.com/p/jinja

[11] https://palletsprojects.com/p/werkzeug



ltdemos.informatik.uni-hamburg.de/factcheck/

Fake New Detection & Enchancing Disinformation...

Specify a claim to verify using additional supporting information

© Soniya Vijayakumar (LT-UHH) 2020. All Rights Reserved.

Figure 6.14: Screenshot of homepage of Fake News Detection Framework

Chapter 7

Evaluation

The major objective of using deep learning models for the respective tasks of veracity classification and justification generation is to leverage transfer learning. To understand the effectiveness of these models in their domain, we perform various quantitative and qualitative analyses. We use the evaluation metrics such as *loss*, *accuracy*, *F1 score*, and *ROUGE* score as explained in Section 3.6. At the beginning of this chapter, we explain the evaluation and comparison studies performed on the deep learning models used in our framework. The later sections illustrate the details of the user study. This chapter concludes by describing the collaboration and feedback we had with a journalist organization.

Model Evaluation

Three deep learning models: **BERT**, **T5**, and **GPT-2** are used in our framework for veracity classification and justification generation tasks respectively. The fine-tuning time for T5 and GPT-2 models are approximately 20 hours (30 epochs) and 8 hours (10 epochs), respectively. This indicates that these models require heavy computational power and fine-tuning is a time-consuming process.

7.1 Veracity Classifier Performance Study

It is well-known that using pre-trained deep learning models enables transfer learning since these models are trained on large datasets and contain extracted features useful for downstream tasks. We follow this approach by fine-tuning BERT for the veracity classification described in Section 6.4. To better understand the performance of the model, we conduct various experiments by fine-tuning the model on different datasets. Three experiment settings are created and each of them uses a different dataset. The model '*bert-base-uncased*' and its architecture remains constant in all the experiments (refer architecture in Section 6.4). The following are the experiment setups and their respective datasets:

1. **Experiment 1: LIAR-PLUS + EUvsDisInfo:** The training and val-

idation dataset is the combination of LIAR-PLUS and 5000 entries from EUvsDisInfo (50%).

2. **Experiment 2: LIAR-PLUS (train + val):** The training and validation dataset is the combination of LIAR-PLUS training and validation set.
3. **Experiment 3: True Fake:** The training and validation dataset is only 60% of TrueFake Dataset.

For understanding the performance of the fine-tuned model in each experiment setup, the test set used is 40% of the TrueFake dataset. Table 7.1 shows the *loss*, *accuracy*, and *F1 scores* on each experiment setup. We also used the pre-trained version of BERT to understand how the model produces test results without fine-tuning. We observe the following (highlighted values indicate the respective *F1 scores*):

1. In the first case, where the model is trained in under experiment 1 setting, we observe a low *F1 score* as the labels are not balanced in the dataset. The EUvsDisInfo adds only fake class claims into the training set. The content of the LIAR-PLUS dataset is real-world news claims whereas EUvsDisInfo contains claims from real-news sources but are more focused on specific news topics (Russia and European Union). The imbalanced label and the difference in content difference could be the contributing factors for a low *F1 score.*

2. In the second case, where the model is trained in experiment setting 2, the *F1 score* improves compared to the first case because the training set and the test set are from the same dataset. The *F1 score* is not as high as the third case, as there is a label balance and this is a contributing factor to the *F1 score.*

3. The highest *F1 score* is observed in the third case, where the model is trained on the TrueFake training and validation dataset and tested on the TrueFake test dataset (experiment 3 setting). This indicates that the model learns representations from the news article content that is scraped from the defined sources. Even though noise data was added to the training dataset, it shows an insignificant effect during the test. Such a high *F1 score* can also be an indication of model overfitting.

A possible way to reduce this overfitting is to introduce more noisy data into the training dataset. Currently, only over a thousand entries are included as noisy data to the golden dataset, which is minimal in size compared to the size of the entire dataset. Another way could be to mix benchmark datasets like FEVER, MultiFC, FakeNewsNet which consists of synthetic and real news content. Thus expanding the scope of the training dataset content.

For experiment 3, we also manually verified the test set to understand the predictions made, in most of the cases the model accurately predicted the classification label, thus indicating this high score. Furthermore, to understand the model

Evaluation	**Training**			**Validation**			**Test: TrueFake Test Set**		
Training Dataset	**Loss**	**Accuracy**	**F1 score**	**Loss**	**Accuracy**	**F1 score**	**Loss**	**Accuracy**	**F1 score**
BERT	-	-	-	-	-	-	0.698	0.486	-
L + E	0.500	0.691	0.683	0.496	0.689	0.672	1.084	0.504	**0.591**
L	0.687	0.546	0.488	0.690	0.516	0.508	0.659	0.679	**0.744**
TF	0.146	0.949	0.950	0.125	0.955	0.956	0.140	0.952	**0.953**

Table 7.1: BERT model performance comparison by fine-tuning on different datasets. The datasets refers to (a) BERT: BERT base (pre-trained), (b) L + E: LIAR-PLUS + EUvsDisInfo, (c) L: LIAR-PLUS (train + val), and (d) TF: TrueFake (60%)

performance concerning fine-tuning, we evaluated the raw pre-trained model. As expected, it has the least accuracy score and this indicates the need for fine-tuning these pre-trained models for domain-specific tasks.

To further understand the performance of our differently fine-tuned models, we studied the evaluation of the above-mentioned experiment setups on different test sets. Table 7.2 shows the results of this evaluation. The following are the observations made:

1. The highlighted values indicate the expected *F1 scores* except in the second case. This is similar to the observation we made in experiment 2 in the previous observations and the same balanced labels could be the contributing factor.

2. In the respective other cases, where the model is trained on a dataset and tested using another dataset, it is evident that the *F1 scores* are lower as there is a difference in the train and test dataset content.

Evaluation	**EUvsDisInfo**		**LIAR-PLUS**		**TrueFake**		**Class Distribution**	
	Accuracy	**F1 score**	**Accuracy**	**F1 score**	**Accuracy**	**F1 score**	**Fake**	**True**
BERT	1.0	1.0	0.436	0.608	0.486	-	-	-
L + E	0.999	**0.999**	0.552	0.115	0.504	0.591	10104 (61.2%)	6420 (38.8%)
L	0.583	0.737	0.560	**0.548**	0.679	0.744	5104 (44.3%)	6420 (55.7%)
TF	0.589	0.742	0.506	0.497	0.952	**0.953**	14342 (51.7%)	13374 (48.3%)

Table 7.2: BERT model evaluation using different training and test datasets. The datasets refers to (a) BERT: BERT base (pre-trained), (b) L + E: LIAR-PLUS + EUvsDisInfo, (c) L: LIAR-PLUS (train + val), and (d) TF: TrueFake (60%)

With these evaluation studies, we observe that the model that is trained in experiment setting 3 has the best *F1 score* and we use this as the pre-loaded veracity classifier in the live demo of our framework. We benefit from transfer learning by fine-tuning the base BERT model using the domain-specific TrueFake dataset. From our user study and qualitative analysis of framework outputs, we observe this model performs fairly well, thus indicating a higher accuracy baseline than the one mentioned in research question 3 (see Section 1.2).

7.2 Justification Generator Performance and Comparison Study

We explored two well-studied natural language processing approached, text summarization and text generation as a method for generating justifications. To understand the performance, we performed automatic quantitative evaluation using different *ROUGE* scores. Table 7.3 shows the various results. T5 is fine-tuned using the justifications present in the LIAR-PLUS dataset whereas GPT-2 is fine-tuned using the justifications present in the EUvsDisInfo dataset.

Approach	**Model**	**ROUGE-1**	**ROUGE-L**	**ROUGELSUM**
Summarization	**Pretrained T5**	0.900	0.866	0.866
	Fine-Tuned T5	0.985	0.985	**0.985**
Text Generation	**Pretrained GPT2**	0.413	0.366	0.338
	Fine-Tuned GPT2	0.471	0.445	0.445

Table 7.3: ROUGE scores for text summarization using T5 model and text generation using GPT-2 model

The pre-trained and fine-tuned models are evaluated using the LIAR-PLUS test set for comparison. It is observed that the highest *ROUGELSUM* score is from the fine-tuned T5 summarization model. The reason for this high score could be overfitting as the train and test set are from the same dataset, hence the model fine-tunes its representations based on the train set and generates summaries using these representations. It is to be noted that, the models were not evaluated using the TrueFake dataset as this dataset does not contain summaries to be used for comparative study.

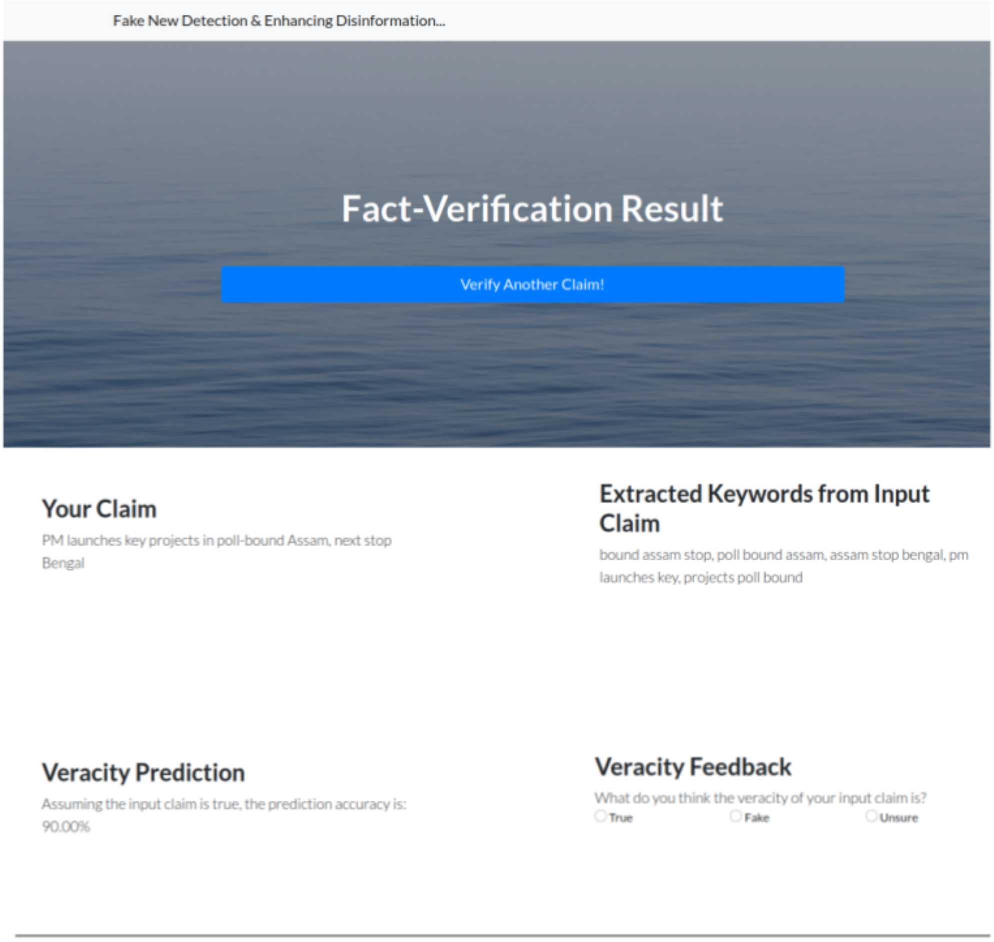
Table 7.4 shows the snippets from fine-tuned T5 text summarization and GPT-2 text generation for the input claim "Royal family in crisis after explosive interview". The summary generated by fine-tuned T5 is fairly relevant and suggestive in nature for the end-user. Whereas, it is important to observe that the fine-tuned GPT-2 model learns topic-relevant words like Russian, disinformation, European Union, which are prominent words in the EUvsDisInfo dataset. This is an evident observation that GPT-2 learns representations specific to the dataset content and hence, indicates the dependency of fine-tuning models on the training dataset.

Summarized Text	Generated Text
the popularity of prince Harry and his wife Meghan has tumbled in Britain over the past year. the hair of the prince is darker than the color of his face. Meghan, duchess of Sussex, bullied royal staff. the duchess Meghan interview was filmed at a friends house. the royal family are "in too big to fail." ohio is number one in the world in terms of heroin overdoses. for the second straight year on the road, tampa bay is running for the top spot in the country. the global warming has killed at least 10 people in the last decade, including british, chinese and russian governments. the prince and the duchess of Sussex are fighting a massive legal battle over baby birmingham. the Sussexes were in a state of extreme poverty. the royal family were not racist. an interview with prince williams father and current gubernatorial candidate george soros showed how much she misses and is untrustworthy. thom tillis is spending his honeymoon in the soviet union	*Royal family in crisis after explosive interview with Stopfake, which was used in Russian disinformation outlets. Royal family in crisis after explosive interview. The family's oldest son died of natural causes from heart failure and cancer. The European Union does not give financial support to a country that participates in the armed conflict in its own right. Royal family in crisis after explosive interview by Reuters*

Table 7.4: Snippets from T5 summarization and GPT-2 generation

7.3 User Study

We conducted a user study by inviting participants to access the framework using the user interface of the Flask based web server. Each participant was sent an instruction guide explaining how to perform fake news detection (see Appendix C). In cases where they wanted to have a veracity reference, they could use fact-checked claims from PolitiFact or other similar sources. The participants also had to enter feedback using the web interface. Examples of the interface are shown in Figures 7.1, 7.2 and 7.3 during the process of detection.



Fact-Verification Result

Your Claim

PM launches key projects in poll-bound Assam, next stop Bengal

Extracted Keywords from Input Claim

bound assam stop, poll bound assam, assam stop bengal, pm launches key, projects poll bound

Veracity Prediction

Assuming the input claim is true, the prediction accuracy is: 90.00%

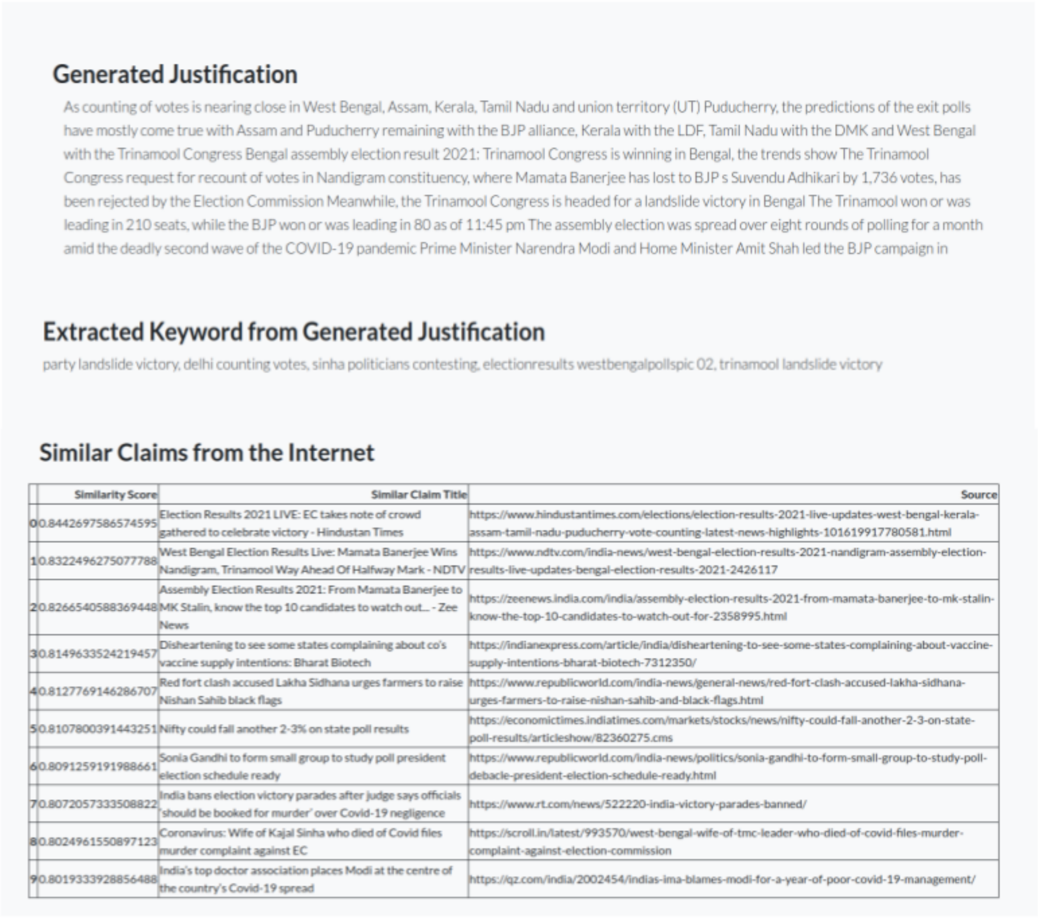
Veracity Feedback

What do you think the veracity of your input claim is?

○ True ○ Fake ○ Unsure

Figure 7.1: Veracity Prediction: User interface outputs from Keyphrase Extractor, Keyphrase Crawler and Veracity Classifier; ‘Journalist-in-the-Loop’: Veracity Feedback

We also gave participants a questionnaire to evaluate the usability of the framework along with written feedback. Figure 7.4 shows the intermediate results from the questionnaire of this user study. It is observed that participants found it straightforward to use the framework but there is mixed feedback on recommend-



Generated Justification

As counting of votes is nearing close in West Bengal, Assam, Kerala, Tamil Nadu and union territory (UT) Puducherry, the predictions of the exit polls have mostly come true with Assam and Puducherry remaining with the BJP alliance, Kerala with the LDF, Tamil Nadu with the DMK and West Bengal with the Trinamool Congress Bengal assembly election result 2021: Trinamool Congress is winning in Bengal, the trends show The Trinamool Congress request for recount of votes in Nandigram constituency, where Mamata Banerjee has lost to BJP s Suvendu Adhikari by 1,736 votes, has been rejected by the Election Commission Meanwhile, the Trinamool Congress is headed for a landslide victory in Bengal The Trinamool won or was leading in 210 seats, while the BJP won or was leading in 80 as of 11:45 pm The assembly election was spread over eight rounds of polling for a month amid the deadly second wave of the COVID-19 pandemic Prime Minister Narendra Modi and Home Minister Amit Shah led the BJP campaign in

Extracted Keyword from Generated Justification

party landslide victory, delhi counting votes, sinha politicians contesting, electionresults westbengalpollspic 02, trinamool landslide victory

Similar Claims from the Internet

	Similarity Score	Similar Claim Title	Source
0	0.8442697586574595	Election Results 2021 LIVE: EC takes note of crowd gathered to celebrate victory - Hindustan Times	https://www.hindustantimes.com/elections/election-results-2021-live-updates-west-bengal-kerala-assam-tamil-nadu-puducherry-vote-counting-latest-news-highlights-101619917780581.html
1	0.8322496275077788	West Bengal Election Results Live: Mamata Banerjee Wins Nandigram, Trinamool Way Ahead Of Halfway Mark - NDTV	https://www.ndtv.com/india-news/west-bengal-election-results-2021-nandigram-assembly-election-results-live-updates-bengal-election-results-2021-2426117
2	0.8266540588369448	Assembly Election Results 2021: From Mamata Banerjee to MK Stalin, know the top 10 candidates to watch out... - Zee News	https://zeenews.india.com/india/assembly-election-results-2021-from-mamata-banerjee-to-mk-stalin-know-the-top-10-candidates-to-watch-out-for-2358995.html
3	0.8149633524219457	Disheartening to see some states complaining about co's vaccine supply intentions: Bharat Biotech	https://indianexpress.com/article/india/disheartening-to-see-some-states-complaining-about-vaccine-supply-intentions-bharat-biotech-7312350/
4	0.8127769146286707	Red fort clash accused Lakha Sidhana urges farmers to raise Nishan Sahib black flags	https://www.republicworld.com/india-news/general-news/red-fort-clash-accused-lakha-sidhana-urges-farmers-to-raise-nishan-sahib-and-black-flags.html
5	0.8107800391443251	Nifty could fall another 2-3% on state poll results	https://economictimes.indiatimes.com/markets/stocks/news/nifty-could-fall-another-2-3-on-state-poll-results/articleshow/82360275.cms
6	0.8091259191988661	Sonia Gandhi to form small group to study poll president election schedule ready	https://www.republicworld.com/india-news/politics/sonia-gandhi-to-form-small-group-to-study-poll-debacle-president-election-schedule-ready.html
7	0.8072057333508822	India bans election victory parades after judge says officials 'should be booked for murder' over Covid-19 negligence	https://www.rt.com/news/522220-india-victory-parades-banned/
8	0.8024961550897123	Coronavirus: Wife of Kajal Sinha who died of Covid files murder complaint against EC	https://scroll.in/latest/993570/west-bengal-wife-of-tmc-leader-who-died-of-covid-files-murder-complaint-against-election-commission
9	0.8019333928856488	India's top doctor association places Modi at the centre of the country's Covid-19 spread	https://qz.com/india/2002454/indias-ima-blames-modi-for-a-year-of-poor-covid-19-management/

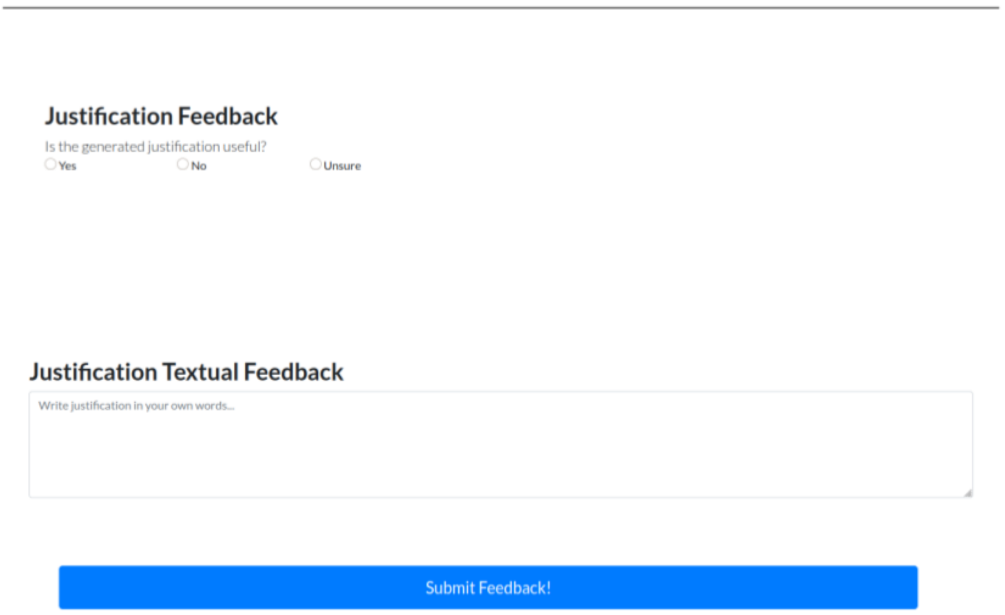
Figure 7.2: Veracity Explanation: User interface outputs from Justification Generator, Keyphrase Extractor and Similar Claim Extractor

ing this system as a **Fake News Detection Framework**. Figure 7.5 shows the textual feedback and the distribution of professional or academic background of the participants. The diversity in professional/academic backgrounds, with users from marketing, business, and computer science professions helped us in understanding the different perspectives of the utility of this framework. Additionally, they also gave us suggestive feedbacks like clarification of the number of data sources and quality of predictions which helped us in qualitative evaluation of the framework and directions for improvement in our future work.

7.4 Journalist Collaboration

During the initial phase and throughout the implementation of the framework, we had the opportunity to collaborate with an organization named CheckFirst[1]. They

[1] https://checkfirst.network/



Justification Feedback

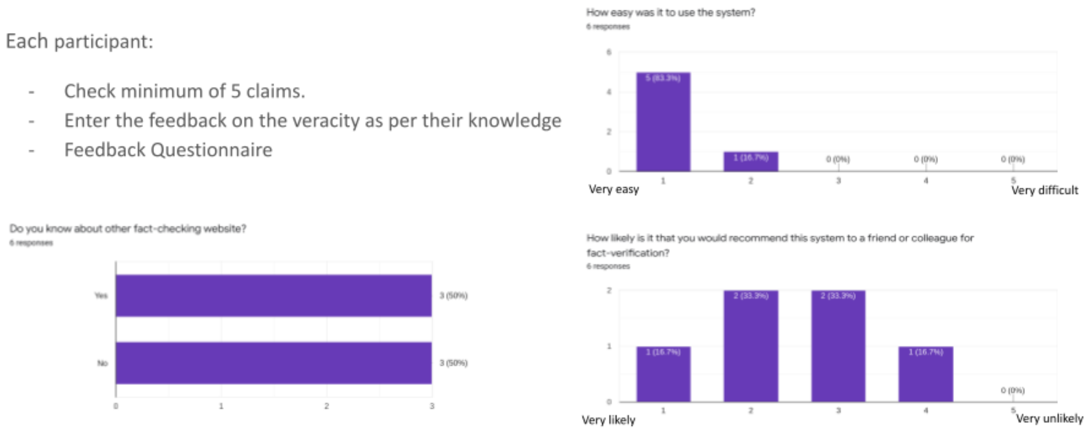
Is the generated justification useful?

○ Yes ○ No ○ Unsure

Justification Textual Feedback

Write justification in your own words...

Figure 7.3: 'Journalist-in-the-Loop': User interface for justification feedback and optional justification textual feedback



Each participant:

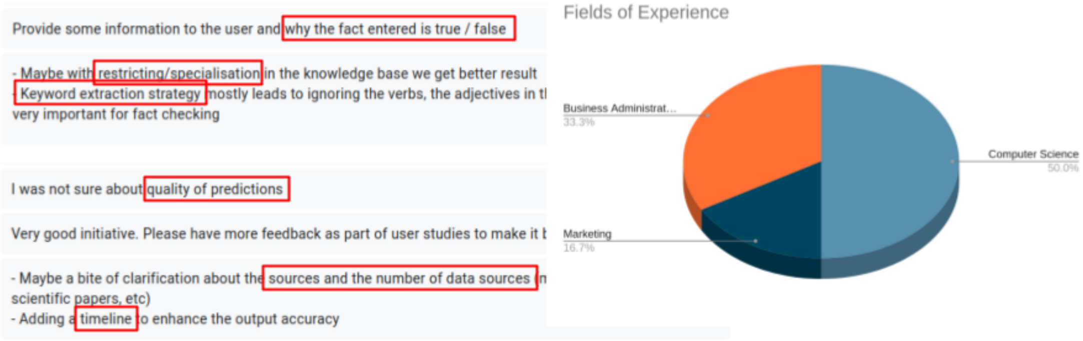
- Check minimum of 5 claims.
- Enter the feedback on the veracity as per their knowledge
- Feedback Questionnaire

Figure 7.4: User study questionnaire statistics

offer software solution tools that help journalists, citizens, and experts connect, organize and debunk fake news. The team is directly involved with end-users who are journalists and is well-aware of the journalistic process of fake news detection. In the earlier phase of our work, they gave us insights into important aspects of the process. After completing the implementation, they also evaluated our framework and gave positive feedback. A few additional directions of research and improvements suggested by them are as follows:

- Explore multi-lingual fact-checking, where a claim might already be fact-checked in another language.
- Semantically related information for the input claim as justification.
- Additional options to show the user, which curated evidence is used for and against the veracity classification.
- Explore additional fact-checking sources while searching for evidence.

Even though each component/module of our framework servers opportunity for further research and enhancements, the overall effectiveness from qualitative and quantitative evaluation shows that the veracity classifier model performs better (95.24% accuracy) on the TrueFake dataset compared to our baseline model performance (50.41%) on LIAR + EUvsDisInfo dataset. In comparison to other relevant existing systems (see Section 2), our framework is the first of its kind to incorporate such elaborate components to constituting an end-to-end framework. The next chapter explains the various challenges we faced while building this end-to-end framework and possible future directions.



Provide some information to the user and why the fact entered is true / false

- Maybe with restricting/specialisation in the knowledge base we get better result
- Keyword extraction strategy mostly leads to ignoring the verbs, the adjectives in tl
very important for fact checking

I was not sure about quality of predictions

Very good initiative. Please have more feedback as part of user studies to make it t

- Maybe a bite of clarification about the sources and the number of data sources (n
scientific papers, etc)
- Adding a timeline to enhance the output accuracy

Figure 7.5: User study questionnaire textual feedback and distribution of participant professional/academic background

Chapter 8

Summary, Conclusion & Future Work

In this chapter, we summarize the framework designed and implemented in this thesis, explain in detail the methodologies we adopted to answer the research questions we formulated (see Section 1.2), and present the various challenges faced during the entire process. We also present future directions that could be addressed and integrated into our framework.

8.1 Summary

In this thesis, we present a novel **Fake News Detection Framework** by including journalists-in-the-loop. The approach involves fine-tuning deep learning models using domain-specific data and leveraging feedback from journalists/end-users to improve the model performance over time. To assist journalists/end-users in their fact-verification system, our framework also generates indicative justifications and extract information from online sources as suggestive evidence. With the task-specific fine-tuning of our baseline model, we achieve an accuracy of 95.2% on the TrueFake test data. The qualitative analysis from our user studies also shows that such a framework is helpful in the fake news detection process and is of contemporary relevance.

The initial step in building such a framework involved creating a knowledge base that collects relevant real-world articles from online news sources. We refer to this knowledge base as **Comprehensive Knowledge Base** (CompKB). It is a continuously growing knowledge base, as it is embedded with the ability to continuously monitor online news sources and create persisted collections of news articles post-processing them. CompKB consists of naturally occurring claims from the internet, in the English language and is realized using a crawler architecture (see Section 4.1). This knowledge base presented could form the baseline for further research directions like determining the trends of fake news topics, or as time advances, which topics dominate the fake news as well as how long they propagate in the online network. Using this knowledge base as the baseline, we extract the dataset for our veracity classifier model.

Our framework is centered around fine-tuning deep learning models using domain - specific datasets. We use three datasets, TrueFake, LIAR-PLUS, and EUvsDisInfo datasets to fine-tune BERT, T5, and GPT-2 transformer models, respectively. The dataset, referred to as TrueFake dataset, is curated by us from two sources: 1) Publicly available, manually verified fake and real news dataset (Ahmed et al., 2018, 2017) and, 2) Subset of CompKB, consisting of online news articles scraped during January 2021. Thus created dataset is our baseline dataset for training the veracity classifier model BERT. For generating explanations, we experiment using two approaches: text summarization and text generation. The LIAR-PLUS dataset, which is an extension of the LIAR dataset with additional full-text verdict reports from PolitiFact, is used to fine-tune the T5 transformer model for our summarization approach. The EUvsDisInfo dataset, a manually curated dataset by one of the dataset presenters at the CLARIN hackathon[1], is used for fine-tuning the text generation model GPT-2 for the text generation approach. This dataset consists of entries that correspond to the ongoing disinformation campaigns of the Russian Federation, along with justifications by the author as to why is it disinformation, the web link, and original language justifications.

We achieve our objective of fake news detection by designing and implementing the **Fake News Detection** framework based on three key pillars: Veracity Prediction, Veracity Explanation, and Journalists-in-the-Loop. Each component, along with additional modules, plays a vital role in detecting the veracity of the input claim, extracting indicative evidence as well as allowing end-users to include their feedback. A Flask-based user interface allows end-users to use this framework in our live demo setup.

Veracity Prediction embeds one of the core objectives of the framework: to determine the veracity of the input claim. To achieve this objective, we fine-tuned the BERT transformer model using the task-specific dataset, the TrueFake Dataset (see Section 6.4). The fine-tuned model is then pre-loaded within the framework to be used as the **Veracity Classifier** model. When the user submits an input claim through the user interface, an evidence set is collated using **Keyphrase Crawler** and **Similar Claim Extractor** modules of the **Veracity Prediction** component. The **Veracity Classifier** model uses this evidence set to predict the veracity of the claim and produces a percentage-based prediction accuracy. We follow this approach, instead of binary classification intending to provide the end-user with assistive information and not a final verdict for the veracity of the claim.

Generating justifications is one of the core steps involved in the journalistic process of fact-verification (Graves, 2017). With this as the foundation, we built the **Veracity Explanation** component as the second pillar of our framework (see Section 6.5). We experiment with two approaches: text summarization and text generation and conduct quantitative and qualitative studies. Based on these studies, we observe that summarization using fine-tuned T5 model generates better justifications compared to those generated by fine-tuned GPT-2 model. We

[1] `https://www.clarin.eu/event/2020/hackathon-covid-19-related-disinformation`

pre-load our framework with the fine-tuned T5 model and use it to generate summaries as justification in our live system. Using these generated justifications, the **Keyphrase Extractor** and **Similar Claim Extractor** modules retrieve the relevant online news articles. The top-K similar articles to the input claim are presented to the end-user as indicative evidence, through the user interface.

We emphasize the importance of feedback from the end-users/journalists and this is encompassed in the third pillar of the framework: **Journalists-in-the-Loop**. This component allows to collect feedback through the user interface from end-users and helps us to create an end-user inclusive framework (see Section 6.6). We ask the end-user about the veracity of the input claim that they checked for, as per their knowledge. We also collect feedback regarding the usefulness of the generated explanation. As more elaborate feedback, we additionally provide optional textual feedback, where they can manually write the justification for the claim in their own words, similar to the manual fact-checking process followed by fact-checkers/journalists.

Referring to the research questions on which the framework is built, we observe that the architectural design and implementation approach followed while building this framework has given good results from qualitative and quantitative evaluations. This indicates that this architecture is an effective baseline for an assistive system for journalists (**RQ1**, see Section 1.2). Regarding **RQ2**, even though this framework is a good starting point, numerous studies have to be conducted with journalists as end-users and we also require to closely work with them to understand the important factors for easing the end-user efforts in fact-checking. Furthermore, we establish a new baseline for the accuracy of the veracity classifier by fine-tuning it on the relevant dataset (**RQ3**). Finally, including outputs from each module and a generated justification as explanations, the framework presents the required information in a human-understandable format (**RQ4**).

8.2 Benefits

The relevant related literature recommends the inclusion of end-user feedback in the fact-verification process. We implemented and evaluated this is as a feature within our framework and is available in our live demo system. With this feedback-inclusive feature, we benefit from an end-to-end automatic fake news detection framework. It consists of components that correspond to a general fact-checking process, where a claim is checked for its veracity and the respective evidence is curated as justifications for the veracity. Additionally, this feedback inclusion allows us to evaluate the performance of the veracity classifier and justification generator models from a qualitative perspective. The feedbacks are appended to the CompKB and the models are fine-tuned periodically, hence enhancing the quality of CompKB and the performance of the models with time. Another key feature is the evidence-based veracity prediction, which allows the veracity classifier model to take into account the relevant curated evidence. This feature enables the framework to predict the veracity of new claims from the current news. We follow a percentage-based veracity classification, which presents the prediction accuracy to the end-user. This approach renders our framework as an assistive system to end-user/journalists in their fact-verification process. Finally, the outputs of every component are presented to the end-user through the user interface, allowing them to understand how the framework processed the input claim to predict the veracity and generated the respective evidence.

8.3 Challenges

Starting from the initial phase to the end, we faced various challenges while building this framework. The most prominent ones are explained below:

Data Collection and Journalist Collaboration: An important aspect for fine-tuning deep learning models is the availability of quality datasets. The collection of the human-annotated gold standard dataset is time-consuming and expensive. We were fortunate to have access to EUvsDisInfo dataset from the author and had a discussion session for understanding the dataset. In the later stages, when we wanted to test this framework in the wild, we were only able to collaborate with an organization that works with journalists. Even though this collaboration gave us valuable insights and feedback into the journalistic process, receiving direct feedback from journalists would be more productive.

Training Artifacts: A known artifact that exists in sequence-to-sequence text generation is the *exposure bias* (Schmidt, 2019). This problem refers to training-inference discrepancy caused in maximum likelihood estimation (MLE) training for auto-regressive neural network language models. This bias is introduced in transformers as they consist of auto-regressive decoders. Even though we have not explicitly tested for the existence of this bias, since the deep learning models used in our framework are transformer-based, such a bias can occur. If such an exposure bias exists in our framework, the results generated by the models during testing/predictions may not be accurate as the model lacks generalization capability. A solution that is proposed by Mihaylova and Martins (2019) is to use a scheduled sampling strategy during training time.

Feedback Artifacts: One of the key contributions of this framework is the inclusion of feedback from end-users. In a scenario where this framework is deployed in a real-time environment like a news organization, this feedbacks can introduce biases into the models over a while. Since the **Veracity Classification** model is re-trained periodically on datasets from **CompKB** that are inclusive of feedbacks, the subjective nature of these feedbacks will influence the performance of the models. A similar challenge is introduced in the text generation approach of the **Justification Generator** model as well, where the style and content of text generation can be influences by the textual feedbacks by the journalists. A possible way to mitigate this bias is to introduce post-processing of these feedbacks using the framework prediction and explanations as reference or by introducing semantic text understanding techniques.

Web Crawling Challenges: There were several crawling issues faced while realizing our **Keyphrase Crawler** module. Most of the time the articles could not be downloaded as it stated that the URL was not found, even though the actual URL, when accessed from a browser was available. We assume these issues are faced due to the security protection of the websites, with SSL/TLS and timeout protocols. For certain sources like NBCNews, the RSS feeds could not be extracted and hence the HTML extraction complemented this issue and vice versa. Content that was not accessible by the Newspaper3K library was not downloaded and hence, even though an upper bound of hundred articles per day/source is defined, there are only a lesser number of articles scraped.

8.4 Future Work

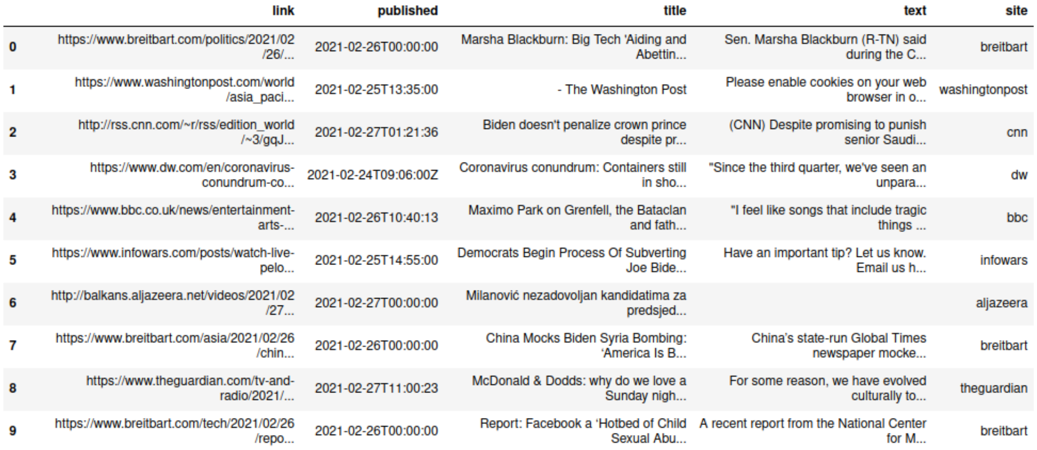
One of the major contributions of this framework is the inclusion of end-user feedbacks and this approach requires to be tested in real-time scenarios to understand the effectiveness and utility of such a framework. Furthermore, by presenting an end-to-end framework for fake news detection, we provide numerous opportunities to enhance each component/module within the fact-verification domain. Various possible directions can be further explored using our framework as a baseline. Currently, our models scrape articles from online news sources from all domains. This search could be narrowed either by searching only for scientific papers, literature books, and journals or by reducing the domain to specific ones like healthcare, major events like elections, life of famous personalities, pandemics, natural disasters. This could also be added as filter options in the user interface and consider these options as per user choice when the **Keyphrase Crawler** searches the web for relevant evidence. From the discussions with CheckFirst[2], we understand that journalists usually explore multi-lingual fact-checking websites and if a claim is already fact-checked in another language, they use this in their verification process. Such a multi-linguistic approach could be embedded into our framework, thus enhancing the search functionality. While using deep learning models, an approach that could be used is an ensemble of transformer models and use the best predictions from them for veracity prediction. This allows us to utilize not just one but multiple state-of-the-art transformer models for our tasks. It is also important to have a manually-annotated golden dataset for fine-tuning these models. Even though the process of manual annotation of data is expensive and time-consuming, it is one of the crucial future requirements. The various challenges mentioned above are also required to be addressed in future enhancements of the framework.

[2] https://checkfirst.network/

Appendices

Appendix A: Framework Outputs

Keyphrase Crawler: A snippet of the Dataframe created while web crawling (see Section 4.1) is shown in Figure 8.1.



	link	published	title	text	site
0	https://www.breitbart.com/politics/2021/02/26/...	2021-02-26T00:00:00	Marsha Blackburn: Big Tech 'Aiding and Abettin...	Sen. Marsha Blackburn (R-TN) said during the C...	breitbart
1	https://www.washingtonpost.com/world/asia_paci...	2021-02-25T13:35:00	- The Washington Post	Please enable cookies on your web browser in o...	washingtonpost
2	http://rss.cnn.com/~r/rss/edition_world/~3/gqJ...	2021-02-27T01:21:36	Biden doesn't penalize crown prince despite pr...	(CNN) Despite promising to punish senior Saudi...	cnn
3	https://www.dw.com/en/coronavirus-conundrum-co...	2021-02-24T09:06:00Z	Coronavirus conundrum: Containers still in sho...	"Since the third quarter, we've seen an unpara...	dw
4	https://www.bbc.co.uk/news/entertainment-arts-...	2021-02-26T10:40:13	Maximo Park on Grenfell, the Bataclan and fath...	"I feel like songs that include tragic things ...	bbc
5	https://www.infowars.com/posts/watch-live-pelo...	2021-02-25T14:55:00	Democrats Begin Process Of Subverting Joe Bide...	Have an important tip? Let us know. Email us h...	infowars
6	http://balkans.aljazeera.net/videos/2021/02/27...	2021-02-27T00:00:00	Milanović nezadovoljan kandidatima za predsjed...		aljazeera
7	https://www.breitbart.com/asia/2021/02/26/chin...	2021-02-26T00:00:00	China Mocks Biden Syria Bombing: 'America Is B...	China's state-run Global Times newspaper mocke...	breitbart
8	https://www.theguardian.com/tv-and-radio/2021/...	2021-02-27T11:00:23	McDonald & Dodds: why do we love a Sunday nigh...	For some reason, we have evolved culturally to...	theguardian
9	https://www.breitbart.com/tech/2021/02/26/repo...	2021-02-26T00:00:00	Report: Facebook a 'Hotbed of Child Sexual Abu...	A recent report from the National Center for M...	breitbart

Figure 8.1: Comprehensive Knowledge Base (CompKB) snippet

Similar Claim Extractor: A detailed example of the most similar and the least similar evidence is shown in table 8.1 along with their similarity scores and the respective articles. The input claim is *'PM launches key projects in poll-bound Assam, next stop Bengal'*

Similarity Score	Article Text
0.8443	**Election Result updates: Rahul Gandhi says he's happy to congratulate Mamata** After a high-octane election campaign in West Bengal, Assam, Tamil Nadu, Kerala, and the union territory of Puducherry, counting of votes has begun. Results for all four states and the UT are expected to pour in from 5 pm. As counting of votes is nearing close in West Bengal, Assam, Kerala, Tamil Nadu and union territory (UT) Puducherry, the predictions of the exit polls have mostly come true with Assam and Puducherry remaining with the BJP alliance, Kerala with the LDF, Tamil Nadu with the DMK and West Bengal with the Trinamool Congress. Prime Minister Narendra Modi has congratulated the winning party chiefs and thanked voters. The counting of votes is being conducted amid tight security in view of the coronavirus disease (Covid-19) and the results are expected to start rolling out after 5pm.
0.7907	**Dissent within the Congress might get louder after poll debacle** For the Congress it brings home the imperative of setting its house in order regardless of who's at the helm in the party. The Congress has yet again managed what it has been doing with nerve-wracking continuity—lose elections! Its comeuppance is heightened by the records its rivals set while beating back its late thrusts in Kerala and Assam. In the states that witnessed direct fights, the Left Democratic Front and the Bharatiya Janata Party broke decades-old hoodoos to retain power. If in Kerala the Marxists-led LDF became the first political formation to win back-to-back, in Assam the BJP is the first non-Congress party to buck anti-incumbency.

Table 8.1: Similarity Score with Respective Article

Appendix B: CompKB Dataset Analysis

Sentiment analysis of real and fake news article text and its headlines are shown in Figures 8.2 and 8.3, respectively. The sentiment analysis is achieved by using the Python 3 TextBlob library[3]. It provides an API and the sentiment property returns a named tuple consisting of polarity and subjectivity. The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

The number of words in real and fake news article headlines is shown in Figure 8.4. Figure 8.5 and 8.6 shows the top 10 positive and negative headlines from the CompKB.

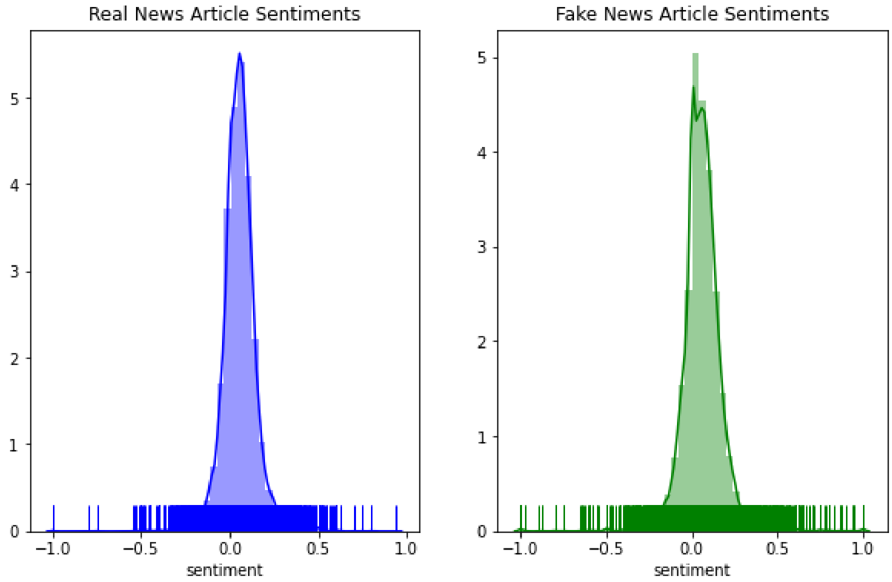


Figure 8.2: Real vs fake news articles sentiment analysis

[3] https://textblob.readthedocs.io/en/dev/

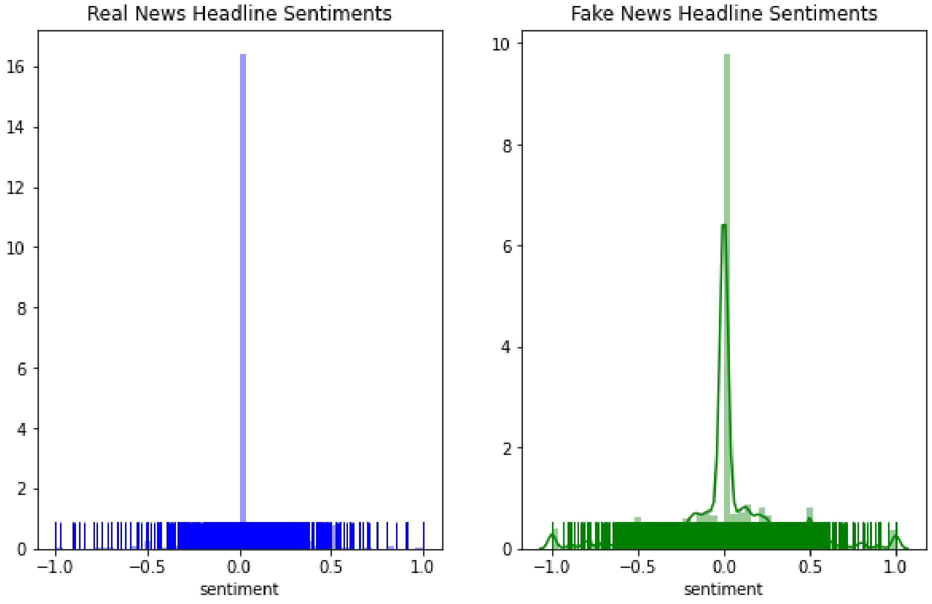


Figure 8.3: Real vs fake news headlines sentiment analysis

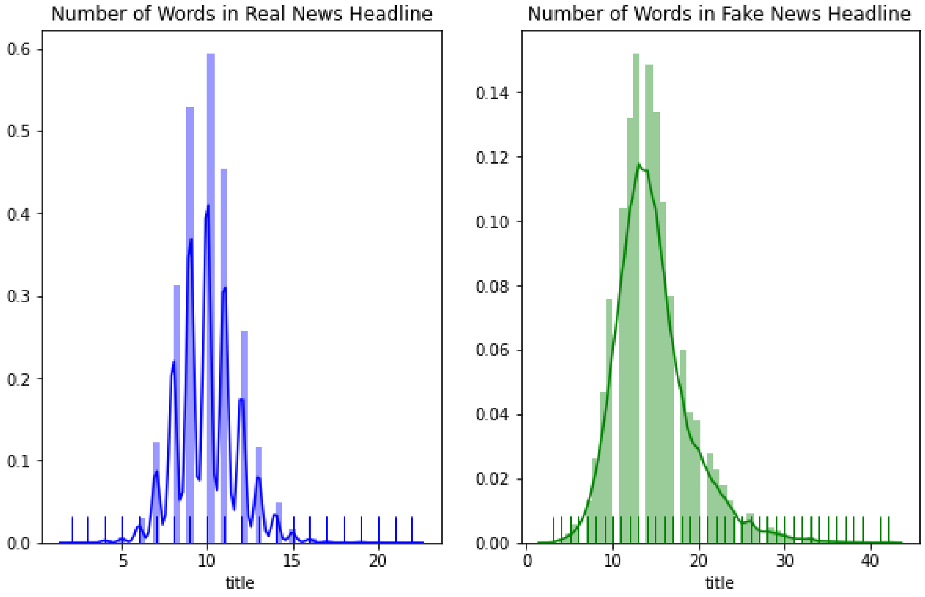
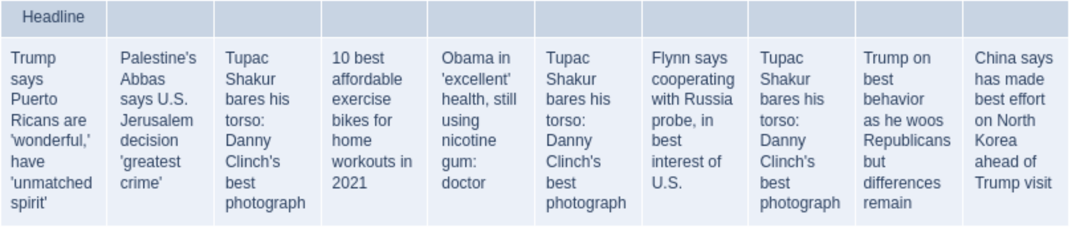
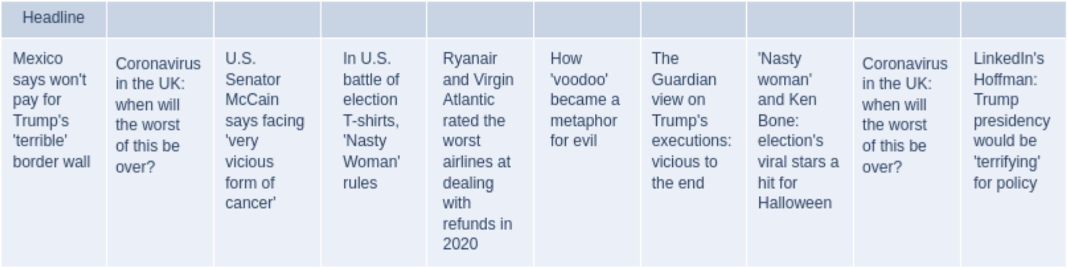


Figure 8.4: Number of words in real vs fake news headlines



Headline									
Trump says Puerto Ricans are 'wonderful,' have 'unmatched spirit'	Palestine's Abbas says U.S. Jerusalem decision 'greatest crime'	Tupac Shakur bares his torso: Danny Clinch's best photograph	10 best affordable exercise bikes for home workouts in 2021	Obama in 'excellent' health, still using nicotine gum: doctor	Tupac Shakur bares his torso: Danny Clinch's best photograph	Flynn says cooperating with Russia probe, in best interest of U.S.	Tupac Shakur bares his torso: Danny Clinch's best photograph	Trump on best behavior as he woos Republicans but differences remain	China says has made best effort on North Korea ahead of Trump visit

Figure 8.5: Top 10 positive headlines



Headline									
Mexico says won't pay for Trump's 'terrible' border wall	Coronavirus in the UK: when will the worst of this be over?	U.S. Senator McCain says facing 'very vicious form of cancer'	In U.S. battle of election T-shirts, 'Nasty Woman' rules	Ryanair and Virgin Atlantic rated the worst airlines at dealing with refunds in 2020	How 'voodoo' became a metaphor for evil	The Guardian view on Trump's executions: vicious to the end	'Nasty woman' and Ken Bone: election's viral stars a hit for Halloween	Coronavirus in the UK: when will the worst of this be over?	LinkedIn's Hoffman: Trump presidency would be 'terrifying' for policy

Figure 8.6: Top 10 negative headlines

Appendix C: User Study Guide

The following are the guidelines for the using the **Fake News Detection Framework**.

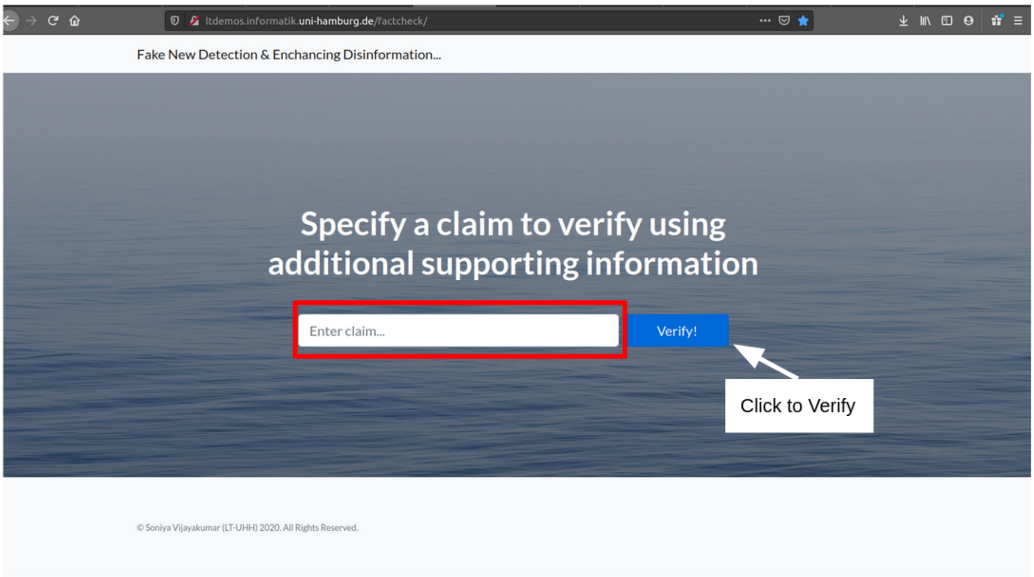
Preface: With an effort to create an assistive system to end-users, we have created this system, for enhancing the process of verifying claims with supportive information. We use deep learning models in this process for predicting the veracity of your input as well as generate justifications for further evidence retrieval from the internet.

URL Access

The system can be accessed in the following link:
http://ltdemos.informatik.uni-hamburg.de/factcheck/

Home Page

Once you open the above link, the following home page will appear:


ltdemos.informatik.uni-hamburg.de/factcheck/

Fake New Detection & Enchancing Disinformation...

Specify a claim to verify using additional supporting information

© Soniya Vijayakumar (LT-UHH) 2020. All Rights Reserved.

Figure 8.7: Fake News Detection Framework user interface homepage

In the text field, highlighted in the above image, enter a 'statement' that you want to verify and click on the 'Verify!' button.

Tip:

- The statement entered will be referred to as 'Claim' in the system

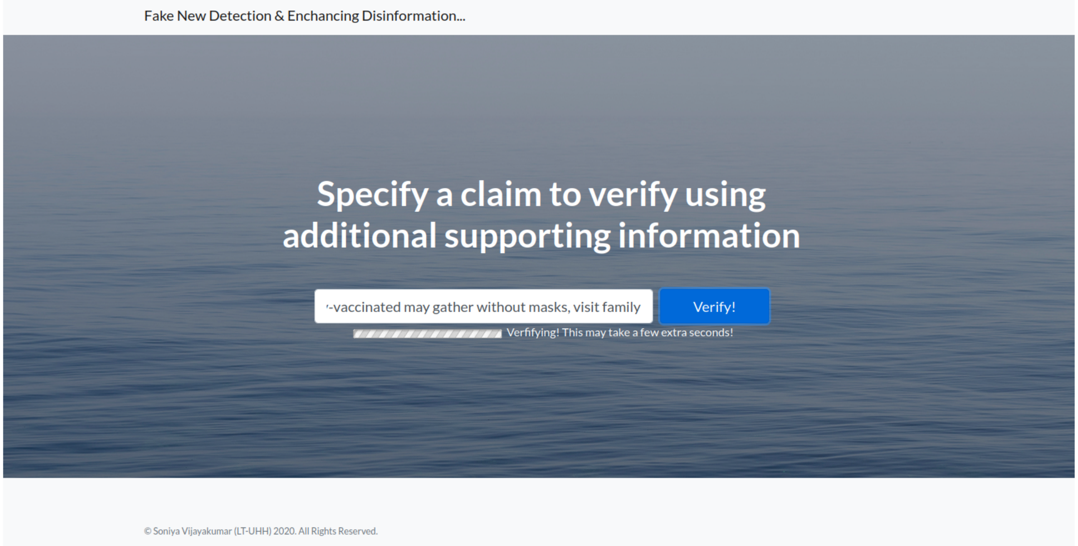
- A few places that you could find a statement to verify are:

  https://www.politifact.com/, https://www.gossipcop.com/

- If you choose a statement from PolitiFact, this site gives you a classification of the statement into the following categories: 'True', 'Mostly True', 'Half True', 'Mostly False', 'False', and 'Pants on Fire'. This will help you have an idea about the veracity of the chosen statement.

- For this user study, please verify at least 5 statements of your choice.

Once the 'Verify!' button is clicked, the following screen is visible to you. The system will take a little while to verify. (While the system is verifying, have a quick stretch to keep yourself active during long periods of work at home times :)



Fake New Detection & Enchancing Disinformation...

Specify a claim to verify using additional supporting information

'-vaccinated may gather without masks, visit family

Verfifying! This may take a few extra seconds!

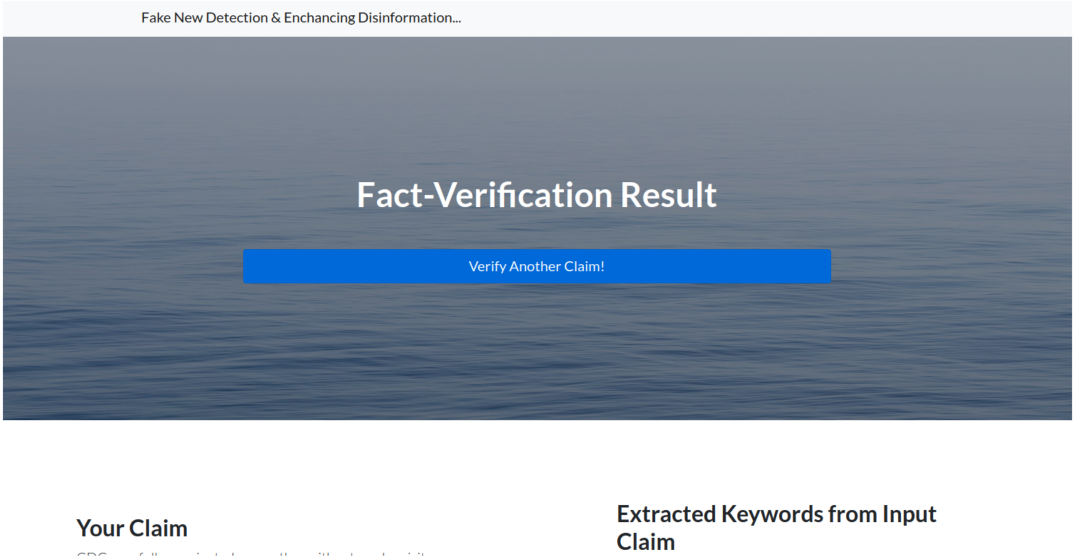
© Soniya Vijayakumar (LT-UHH) 2020. All Rights Reserved.

Figure 8.8: Fake News Detection Framework user interface homepage

Once the system verifies your input, the result is displayed. Refer next section for detailed explanation of the results

Fake News Detection Results

The result page is as below:



Fake New Detection & Enchancing Disinformation...

Fact-Verification Result

Your Claim

Extracted Keywords from Input Claim

Figure 8.9: Fake News Detection Framework user interface results page header

Note: Post checking the results, you may verify further claims by clicking on the 'Verify Another Claim!' button.

The result page consists of sections as shown in Figures 8.10, 8.11 and 8.12.

Section: Your Claim

This section displays the 'statement' that you entered, which will be referred to as 'claim', henceforth.

Section: Extracted Keyphrases from Input Claim

Keywords are extracted from your input claim to search for relevant information, which is then used by the veracity prediction module to predict input claim's veracity. These keywords are displayed here.

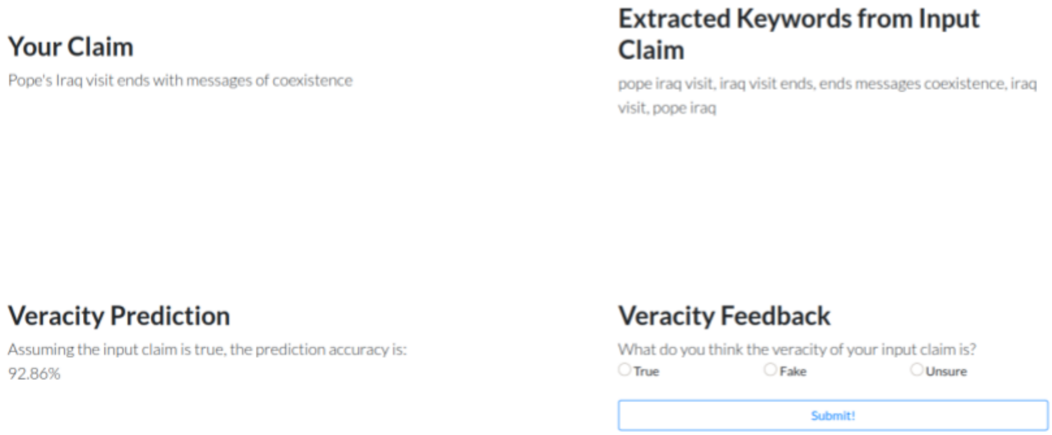
Section: Veracity Prediction

Assuming that your input claim is true, this section displays the prediction accuracy. As shown in the example, the prediction accuracy of 92.86% means that your input claim is true with a confidence of 92.86%. In other words, the model predicts the claim to be false with 7.14% confidence.

Section: Veracity Feedback

This section allows you to enter your thoughts about the veracity of the input claim. If you think or know that your input claim is true, please select 'True' else

'Fake'. If you are unsure, you can select the "unsure" option. This selection has to be made irrespective of the prediction accuracy in the previous section.



Your Claim

Pope's Iraq visit ends with messages of coexistence

Extracted Keywords from Input Claim

pope iraq visit, iraq visit ends, ends messages coexistence, iraq visit, pope iraq

Veracity Prediction

Assuming the input claim is true, the prediction accuracy is: 92.86%

Veracity Feedback

What do you think the veracity of your input claim is?

○ True ○ Fake ○ Unsure

Figure 8.10: Fake News Detection Framework user interface results page veracity prediction section

Section: Generated Justification

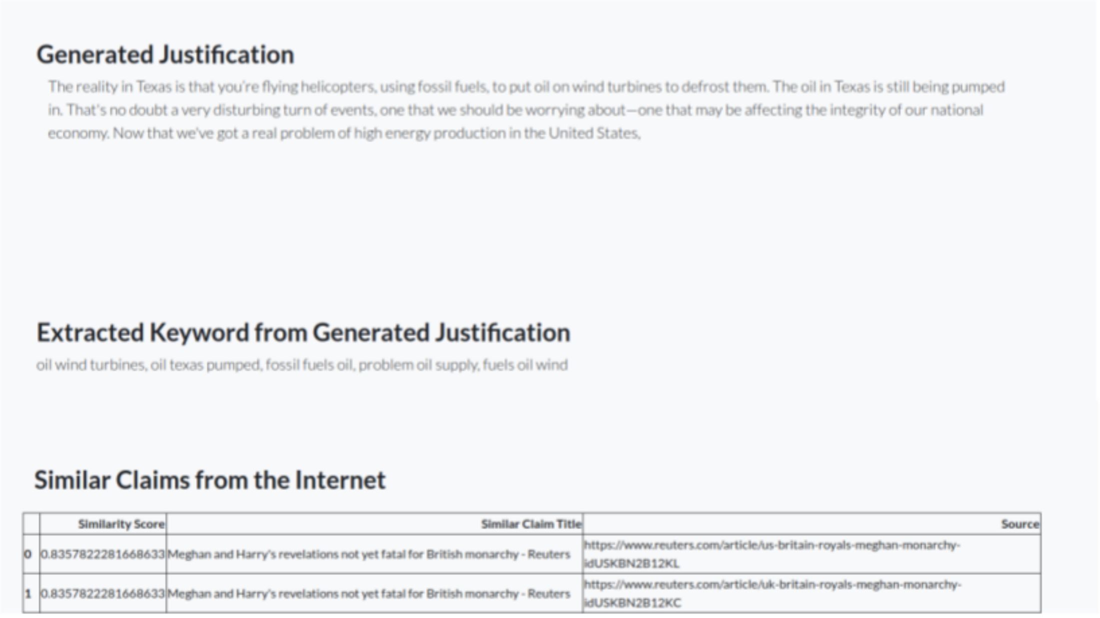
A justification is generated based on your input claim and the relevant information gathered, in order to assist in creating a justification for the claim's veracity. This generated justification is displayed in this section. This justification is purely meant for assisting end-users in writing justifications regarding the input claim.

Section: Extracted Keyphrases from Generated Justification

In order to further assist in creating the justification, we extract keywords from the generated justification. The same is displayed in this section.

Section: Similar Claims from the Internet

The relevant information related to the generated justification and your input claim from the internet is displayed in this section. The top 10 most similar results are displayed. The similarity score in each row indicates how similar each row is to your input claim.



Generated Justification

The reality in Texas is that you're flying helicopters, using fossil fuels, to put oil on wind turbines to defrost them. The oil in Texas is still being pumped in. That's no doubt a very disturbing turn of events, one that we should be worrying about—one that may be affecting the integrity of our national economy. Now that we've got a real problem of high energy production in the United States,

Extracted Keyword from Generated Justification

oil wind turbines, oil texas pumped, fossil fuels oil, problem oil supply, fuels oil wind

Similar Claims from the Internet

	Similarity Score	Similar Claim Title	Source
0	0.8357822281668633	Meghan and Harry's revelations not yet fatal for British monarchy - Reuters	https://www.reuters.com/article/us-britain-royals-meghan-monarchy-idUSKBN2B12KL
1	0.8357822281668633	Meghan and Harry's revelations not yet fatal for British monarchy - Reuters	https://www.reuters.com/article/uk-britain-royals-meghan-monarchy-idUSKBN2B12KC

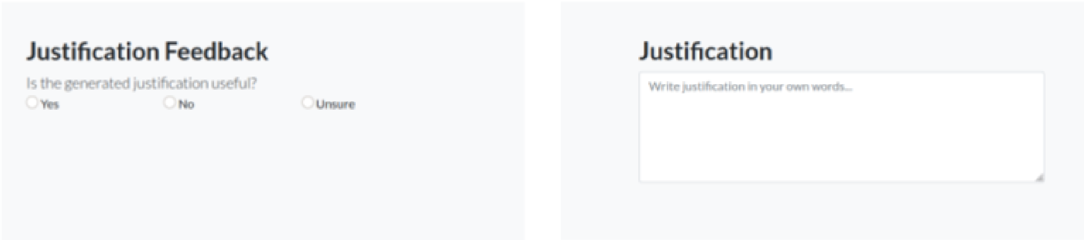
Figure 8.11: Fake News Detection Framework user interface results page veracity explanation section

Section: Justification Feedback

This section allows you to enter your thoughts about usability of the generated justification

Section: Justification

This section allows you to enter justification manually for the input claim that you have checked.



Justification Feedback

Is the generated justification useful?

○ Yes ○ No ○ Unsure

Justification

Write justification in your own words...

Figure 8.12: Fake News Detection Framework user interface results page feedback section

Once you submit your feedback, you will be redirected to the following page and you can verify further claims, as you wish to.

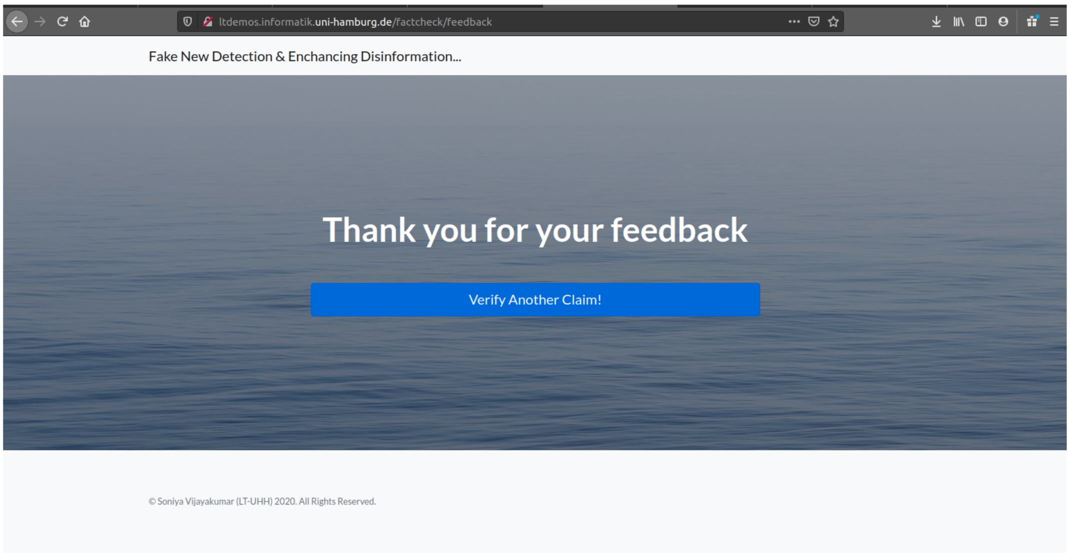

ltdemos.informatik.uni-hamburg.de/factcheck/feedback
Fake New Detection & Enchancing Disinformation...
Thank you for your feedback
Verify Another Claim!
© Soniya Vijayakumar (LT-UHH) 2020. All Rights Reserved.

Figure 8.13: Fake News Detection Framework user interface thankyou page

Thank you for using this system. Please also use this form to give additional feedback.

Appendix D: Claim Verification Example Outputs

This section illustrates outputs generated using the live demo of our framework, with claims chosen from PolitiFact. The link for the demo can be found in this repository.

Example 1

Input Claim: “Says Anthony Fauci will make millions off new book.”

The input claim used here is stated to be false by the PolitiFact Truth-o-Meter.

The following sections show the output that our framework generated as a part of the fake news detection process. The outputs in blue corresponds to the framework generated outputs.

Section: Your Claim

Says Anthony Fauci will make millions off new book.

Section: Extracted keyphrases from Input Claim

millions new book, make millions new, fauci make millions, anthony fauci make, says anthony fauci

The **Keyphrase Extractor** module extracts the above trigrams as keyphrases.

Section: Veracity Prediction

Assuming the input claim is true, the prediction accuracy is: 30.00%

The trigrams extracted by the **Keyphrase Extractor** is used by the **Keyphrase Crawler** to search for relevant article in the online news sources. A set of top-10 evidence are collated by the **Similar Claim Extractor** module and sent to the **Veracity Classifier** module. This module assumes that the input claim is true and predicts its veracity. The 30.00% prediction is inferred as: “Consider the input claim to be true and the classifier predicts the claim to be true by 30%”. This means, if a threshold of 50% is considered, then the prediction says that this claim is false, as the accuracy is below the threshold. Such a percentage-based prediction is given to the end-user as an indicative measure of the truthfulness of the claim.

Section: Generated Justification

The following is the justification generated by the **Justification Generator** module using the summarization approach. The content below is an abstractive summary of the article text from the evidence set used for veracity classification.

Note: The justification is longer than the excerpt shown here. It is snipped out for readability purpose.

The author had worked out a solid strategy for the business book he was writing. He had secured two published authors to contribute to the book, one of whom had sold millions of books himself I was excited about the marketing plan, but something about the book itself didn't "click" for me. Your book needs to solve a problem Too many new authors think of a "mega-concept" for their first book They want to go epic; they want to write a book that answers everything and is so useful that it will have something in it for everybody! Well, that book already exists. The Washington Post obtained 866 pages of Anthony Fauci's emails from March and April 2020 via the Freedom of Information Act, revealing what it was like for the government's top infectious-disease expert to navigate the chaotic early weeks of the pandemic under the Trump administration Why it matters: Fauci has been one of the most prominent public faces of the U officials were eventually forced to assign Fauci a full-time security : Fauci told the Post that he would receive approximately 1,000 emails a day from colleagues, politicians, medical workers, foreign governments and strangers "I was getting every single kind of question, mostly people who were a little bit confused about the mixed messages that were coming out of the White House and wanted to know what's the real scoop," Fauci said Highlights George Gao, director of the Chinese Center for Disease Control and Prevention, emailed Fauci on March 28, shortly after Science magazine quoted Gao saying that the U Anthony Fauci, the chief medical advisor to the President and the director of the National Institute of Allergy and Infectious Diseases, spoke with YouTube's Dr Fauci Warned That Cases Could Rise Again "I think it's very positive," Dr Fauci said when asked about our progress to far, "but one of the things that we get concerned about is declaring victory prematurely Fox News host Tucker Carlson verbally attacked Anthony Fauci on Wednesday and said the director of the National Institute of Allergy and Infectious Diseases should be under criminal investigation The remarks toward Fauci, President Joe Biden's chief medical advisor, came after BuzzFeed and The Washington Post released thousands of his emails from the early months of the COVID-19 pandemic The emails showed Fauci received a theory in January 2020 that the virus could have leaked from a lab in Wuhan, China....

Section: Extracted Keyphrases from Generated Justification

book sold millions, sold millions books, books excited marketing, millions books excited, released thousands emails

Further, the **Keyphrase Extractor** extracts trigrams from the generated justification. This allows to understand the difference in keyphrases that are extracted from the input claim and the generated justification

Section: Similar Claims from the Internet

Using the extracted keyphrases, once again the **Keyphrase Crawler** crawls the online sources for articles. Using top-10 similarity approach, the **Similar Claim Extractor** module collates this as a set of indicative evidence to the input claim and presents it to the end-user in a tabular form. Table 8.2 shows the similarity score of the article to the input claim, the title of the article and the source link for the end-user to read further.

Similarity Score	**Similar Claim Title**	**Source**
0.9072	Avoid This Common Mistake When Writing Your First Business Book	https://www.entrepreneur.com/article/373825
0.9041	Top Stories this PM: 70K sign petitions to keep Bezos in space; Kushner's meltdown over masks	https://www.businessinsider.com/latest-news-70000-people-sign-petitions-to-keep-bezos-in-space-2021-6
0.8928	The Out-of-Touch Adults' Guide To Kid Culture: How Old Is Evan Hansen Again?	https://lifehacker.com/the-out-of-touch-adults-guide-to-kid-culture-how-old-i-1846940794
0.8915	Got the jab, bought the T-shirt: 'vaxinistas' and the rise of pandemic merchandise	https://amp.theguardian.com/fashion/2021/jun/15/got-the-jab-bought-the-t-shirt-vaxinistas-and-the-rise-of-pandemic-merchandise
0.88870	866 pages of Fauci emails shed light on early days of COVID crisis - Axios	https://www.axios.com/anthony-fauci-emails-covid-trump-651d1686-2dc2-4233-b899-cc8135ec8403.html
0.88491	If This Sounds Like You, You're At Risk of COVID, Says Dr. Fauci — Eat This Not That - Eat This, Not That	http://www.eatthis.com/news-fauci-covid-risk-people-warning/
0.88372	Tucker Carlson slams Fauci as 'Jesus for people who don't believe in God' on Fox News	https://news.yahoo.com/tucker-carlson-slams-fauci-jesus-131419280.html
0.8832	Coronavirus latest news: Working from home set to stay as Government has 'no intention' of forcing people back to offices	https://news.yahoo.com/coronavirus-latest-news-almost-30-221537711.html

Table 8.2: Example 1 Similar claim from the Internet as suggestive evidence

As indicative reference of the sources, Table 8.3 shows the sources that is shown in PolitiFact.

Twitter, Charlie Kirk post, June 1, 2021
PolitiFact, social media using old Fauci email falsely claims that Fauci 'lied' about mask wearing, June 2, 2021
PolitiFact, No, emails to Fauci don't show early agreement that virus was man-made, June 2, 2021
Deadline, Dr. Anthony Fauci Book Scrubbed From Amazon, Barnes & Noble After Premature Posts, June 2, 2021
Forbes, Dr. Anthony Fauci: The Highest Paid Employee In The Entire U.S. Federal Government, Jan. 25, 2021
Email interview, Chris Albert spokesperson at National Geographic, June 15, 2021

Table 8.3: Example 1 Sources from PolitiFact

Example 2

Input Claim: "It's been over 50 years since minimum (wage) and inflation parted ways, then over a decade since the federal minimum went up at all."

The input claim used here is stated to be true by the PolitiFact Truth-o-Meter.

The following sections show the output that our framework generated as a part of the fake news detection process. The outputs in blue corresponds to the framework generated outputs.

Section: Your Claim

It's been over 50 years since minimum (wage) and inflation parted ways, then over a decade since the federal minimum went up at all.

Section: Extracted keyphrases from Input Claim

parted ways decade, 50 years minimum, decade federal minimum, ways decade federal, wage inflation parted

The **Keyphrase Extractor** module extracts the above trigrams as keyphrases.

Section: Veracity Prediction

Assuming the input claim is true, the prediction accuracy is: 90.00%

The trigrams extracted by the **Keyphrase Extractor** is used by the **Keyphrase Crawler** to search for relevant article in the online news sources. A set of top-10 evidence are collated by the **Similar Claim Extractor** module and sent to the **Veracity Classifier** module. This module assumes that the input claim is true and predicts its veracity. The 90.00% prediction is inferred as: "Consider the input claim to be true and the classifier predicts the claim to be true by 90%". This means, if a threshold of 50% is considered, then the prediction says that this claim is true, as the accuracy is above the threshold. Such a percentage-based prediction is given to the end-user as an indicative measure of the truthfulness of the claim.

Section: Generated Justification

The following is the justification generated by the **Justification Generator** module using the summarization approach. The content below is an abstractive summary of the article text from the evidence set used for veracity classification.

Note: The justification is longer than the excerpt shown here.

Six hundred billion dollars per year, and growing: That is two-thirds of total nondefense discretionary spending by the federal government, about what is spent on defense operations, military personnel and procurement, and more than mandatory federal expenditures on Medicaid The five of us served as Treasury secretary under three presidents, both Republican and Democrat, representing 17 years of experience at the helm of the department Charles Platiau Data distorted by COVID effects Businesses struggle to pass cost rise to consumers Post-COVID factors could accentuate downward pressures. Faced with shortages of hospitality staff, Australia's Queensland state wants to lure chefs, bartenders and tour guides to its sun-kissed beaches with a "Work In Paradise" scheme of one-off incentives and help with travel costs "Businesses are trying to cope with the (labor) shortage in different ways but we aren't seeing industry-wide wage pressures," said Daniel Gschwind, chief executive of the Queensland Tourism Industry Council In the decade since the global financial crisis, wage growth around the world was struggling to recover even before COVID-19 lockdowns last year pushed it down still further in many countries, according to the International Labour Organization Nowhere is this more evident than in the rising popularity of a Federal Reserve program that lets firms stash their cash overnight with the U Show caption Research shared exclusively with Guardian Money by SunLife found that a quarter of over-50s don't have a private or company pension 12 Research shared exclusively with Guardian Money found that a quarter of over-50s don't have a private or company pension 42 BST Share on Facebook Share on Twitter Share via Email. The G7 group of wealthy nations signed a historic tax agreement to tackle tax abuses by multinationals and online technology companies on Saturday, agreeing to a minimum global corporate tax rate for the first time Although broadly welcomed by tax campaigners and labelled a moment that would "change the world" by G7 finance ministers, months and possibly years of talks still need to take place before the rules come into force Here is what is at stake: What has the G7 agreed? There are two main pillars to the agreed reforms: one enabling countries to tax some of the profits made by big companies based on the revenue they generate in that country, rather than where the firm is located for tax purposes, and a second setting a minimum global corporation tax rate While the home-care industry, which already has high turnover rates, constantly needs new workers, the franchisees described an unprecedented shortage of new caregivers in 2021 s increasingly competitive minimum-wage labor market " 4:13 pm: The Diamondbacks announced this afternoon they've parted ways with hitting coach Darnell Coles and assistant hitting coach Eric Hinske

Section: Extracted Keyphrases from Generated Justification

tax abuses multinationals, decade global financial, presidents republican democrat, global corporate tax, global financial crisis

Further, the **Keyphrase Extractor** extracts trigrams from the generated justification. This allows to understand the difference in keyphrases that are extracted from the input claim and the generated justification

Section: Similar Claims from the Internet

Using the extracted keyphrases, once again the **Keyphrase Crawler** crawls the online sources for articles. Using top-10 similarity approach, the **Similar Claim Extractor** module collates this as a set of indicative evidence to the input claim and presents it to the end-user in a tabular form. Table 8.4 shows the similarity score of the article to the input claim, the title of the article and the source link for the end-user to read further.

Similarity Score	Similar Claim Title	Source
0.9404	Five Former Treasury Secretaries: Fund the IRS	https://www.nytimes.com/2021/06/09/opinion/politics/irs-tax-evasion-geithner-lew-paulson-summers-rubin.html
0.9383	Analysis: In Paradise and beyond, wage hikes lag global recovery - Reuters	https://www.reuters.com/world/the-great-reboot/paradise-beyond-wage-hikes-lag-global-recovery-2021-05-25/
0.9378	Analysis: A 'tsunami' of cash is driving rates ever lower. What will the Fed do? - Reuters	https://www.reuters.com/business/finance/tsunami-cash-is-driving-rates-ever-lower-what-will-fed-do-2021-06-03/
0.9377	Over 50? It's not too late to start saving in a pension	https://amp.theguardian.com/money/2021/jun/05/over-50-its-not-too-late-to-start-saving-in-a-pension
0.9351	FACTBOX-Tackling the wealth gap: governments, central banks step up to act - Reuters	https://www.reuters.com/article/global-economy-inequality-factbox-idUSL5N2NC27T
0.93510	Factbox: Tackling the wealth gap: governments, central banks step up to act - Reuters	https://www.reuters.com/business/tackling-wealth-gap-governments-central-banks-step-up-act-2021-05-26/
0.9337	G7 tax reform: what has been agreed and which companies will it affect?	https://amp.theguardian.com/world/2021/jun/07/g7-tax-reform-what-has-been-agreed-and-which-companies-will-it-affect
0.9313	From guaranteeing full-time work to giving out gas cards, the shorthanded home-care industry is pulling out all the stops to hire more caregivers	https://www.businessinsider.com/home-care-industry-franchises-pulling-out-all-stops-in-hiring-2021-5
0.9310	A 20-Foot Sea Wall? Miami Faces the Hard Choices of Climate Change.	https://www.nytimes.com/2021/06/02/us/miami-fl-seawall-hurricanes.html

Table 8.4: Example 2 Similar claim from the Internet as suggestive evidence

As indicative reference of the sources, Table 8.5 shows the sources that is shown in PolitiFact.

Mandela Barnes tweet, May 23, 2021
Politico, “8 Democrats defect on $15 minimum wage hike,”, March 5, 2021
Milwaukee Journal Sentinel, “Wisconsin budget battle begins: GOP lawmakers plan to remove hundreds of items from Gov. Tony Evers’ proposal,” May 5, 2021
USA TODAY, “$15 minimum wage would boost pay for millions but would cost 1.4 million jobs, report says,” Feb. 8, 2021
H.R. 2, Fair Minimum Wage Act of 2007, accessed June 4, 2021
U.S. Department of Labor, Minimum Wage, accessed June 7, 2021
Forbes, “What you need to know about the minimum wage debate,” Feb. 26, 2021
Economic Policy Institute, Congress has never let the federal minimum wage erode for this long, June 17, 2019
Congressional Research Service, The Federal Minimum Wage: Indexation, Oct. 26, 2016
Khan Academy, Indexation and its limitations, accessed June 11, 2021
U.S. Department of Labor, History of Federal Minimum Wage Rates Under the Fair Labor Standards Act, 1938 - 2009, accessed June 11, 2021

Table 8.5: Example 2 Sources from PolitiFact

Example 3

Input Claim: "It was "the left" that "finally" made Juneteenth a national holiday."

The input claim used here is stated to be Half True by the PolitiFact Truth-o-Meter.

The following sections show the output that our framework generated as a part of the fake news detection process. The outputs in blue corresponds to the framework generated outputs.

Section: Your Claim

It was "the left" that "finally" made Juneteenth a national holiday.

Section: Extracted keyphrases from Input Claim

left finally juneteenth, juneteenth national holiday, finally juneteenth national

The **Keyphrase Extractor** module extracts the above trigrams as keyphrases.

Section: Veracity Prediction

Assuming the input claim is true, the prediction accuracy is: 40.00%

The trigrams extracted by the **Keyphrase Extractor** is used by the **Keyphrase Crawler** to search for relevant article in the online news sources. A set of top-10 evidence are collated by the **Similar Claim Extractor** module and sent to the **Veracity Classifier** module. This module assumes that the input claim is true and predicts its veracity. The 40.00% prediction is inferred as: "Consider the input claim to be true and the classifier predicts the claim to have a veracity of 40%". This is a subjective evaluation as PolitiFact says its half true and our framework says its true by 40%. Such a percentage-based prediction is given to the end-user as an indicative measure of the truthfulness of the claim.

Section: Generated Justification

The following is the justification generated by the **Justification Generator** module using the summarization approach. The content below is an abstractive summary of the article text from the evidence set used for veracity classification.

Note: The justification is longer than the excerpt shown here. It is snipped out for readability purpose.

It's time for a reset : Macy Gray proposes to change American flag on Juneteenth. Juneteenth is now a federal holiday after Biden signs bill into law President Biden signed legislation establishing a new federal holiday commemorating the end of slavery USA TODAY Celebrities are speaking out about the importance of Juneteenth While some have always honored Juneteenth, the Black Lives Matter protest movement against racial injustice last summer in response to the deaths of George Floyd, Breonna Taylor and other Black Americans sparked increased attention for the event, now a federal holiday This week, both chambers of Congress passed the Juneteenth National Independence

Day Act and on Thursday, President Joe Biden signed the bill into law Juneteenth: Biden signs Juneteenth into a holiday, officially giving federal employees the day off Friday More: Juneteenth 2021 celebrations: What to know about the holiday Macy Gray proposes to change American flag Grammy-winning singer Macy Gray suggested the flag could use an update for Juneteenth after she said the meaning of the flag was “hijacked” after the Jan The 94-year-old activist from Fort Worth, Texas, who is oft-referred to as the “Grandmother of Juneteenth,” has already begun her annual Walk to D, as part of her efforts to see the momentous day recognized as a federal holiday Each year on June 19, Lee makes a two-and-a-half-mile pilgrimage to commemorate the date in 1865, two and a half years after Abraham Lincoln’s Emancipation Proclamation, when more than 250,000 enslaved Black people in Texas learned that they were finally free, marking the true end of slavery in America The resulting holiday, Juneteenth — also known as Freedom Day, Jubilee Day, Liberation Day and Emancipation Day — has long been a major celebration in Texas, but until Thursday, when President Biden signed a bill establishing Juneteenth as a federal holiday, not all 50 states recognized or commemorated it. For more than 40 years, she had carried on the tradition, working with the National Juneteenth Observance Foundation and leading local Juneteenth events , to petition the Obama administration and Congress to grant the holiday an official position on the calendar I walked from the church, two and a half miles, went home, and the next day I started where I left off ” From September 2016 to January 2017, Lee traveled the country, marching the symbolic two-and-a-half-mile stretch in cities that invited her to take part in their Juneteenth festivities ” Opal Lee at the National Press Club in Washington, D I’m overwhelmed at the people who didn’t know about Juneteenth and it’s just coming to their attention ” Lee was spurred to preserve the historical significance of the holiday, having grown up in a time not far removed from racial horrors such as the Red Summer of 1919 and the Tulsa Race Massacre of 1921, which many citizens have recently learned about “Recognizing Juneteenth nationally would be one more way to acknowledge the intrinsic value of Black people and their history to the wealth and prosperity of the USA,” Nyong’o tweeted to her 1 Today, President Biden “signed the Juneteenth National Independence Day Act, making Juneteenth a federal holiday,” Jimmy Fallon said on Thursday s Tonight Show) on Tuesday announced he will no longer obstruct efforts to make Juneteenth a federal holiday — and with that, the United States second, fuller Independence Day may finally receive the official recognition long overdue “Throughout history, Juneteenth has been known by many names: Jubilee Day Here are the 14 Republican representatives who voted against making Juneteenth a federal holiday Juneteenth is finally getting some mainstream recognition as a holiday Celebrated on June 19, Juneteenth has sometimes been referred to as America’s “second Independence Day

Section: Extracted Keyphrases from Generated Justification

celebrated juneteenth president, invited juneteenth festivities, federal holiday juneteenth, events petition obama, celebration texas thursday

Further, the **Keyphrase Extractor** extracts trigrams from the generated justification. This allows to understand the difference in keyphrases that are extracted from the input claim and the generated justification

Section: Similar Claims from the Internet

Using the extracted keyphrases, once again the **Keyphrase Crawler** crawls the online sources for articles. Using top-10 similarity approach, the **Similar Claim Extractor** module collates this as a set of indicative evidence to the input claim and presents it to the end-user in a tabular form. Table 8.7 shows the similarity score of the article to the input claim, the title of the article and the source link for the end-user to read further.

Facebook post, June 17, 2021
PolitiFact, "Juneteenth's 156-year path to becoming a federal holiday: Here's the history," June 17, 2021
Congress.gov, "Roll Call 170 Bill Number: S. 475," June 16, 2021
Congress.gov, "S.475 - Juneteenth National Independence Day Act," accessed June 19, 2021
Congress.gov, "H.R.7232 - Juneteenth National Independence Day Act," accessed June 19, 2021
Congress.gov, "S.4019 - Juneteenth National Independence Day Act," accessed June 19, 2021
Sen. Ed Markey, news release, Feb. 25, 2021

Table 8.6: Example 3 Sources from PolitiFact

As indicative reference of the sources, Table 8.6 shows the sources that is shown in PolitiFact.

Similarity Score	**Similar Claim Title**	**Source**
0.9155	Usher attends signing to make Juneteenth a federal holiday: ‘Long overdue’	https://www.usatoday.com/story/entertainment/celebrities/2021/06/18/juneteenth-federal-holiday-celebrities-react-celebrate/7726515002/
0.9088	Trump went on rant about ‘the Blacks’ after protests over George Floyd murder	https://news.yahoo.com/trump-went-rant-blacks-protests-200700237.html
0.9074	Why 94-Year-Old Activist Opal Lee Marched to Make Juneteenth a National Holiday	https://variety.com/2021/politics/features/activist-opal-lee-juneteenth-holiday-1234998507/
0.8997	Jimmy Fallon and Stephen Colbert cheer the new Juneteenth holiday, jeer the 14 congressmen who voted against it	https://news.yahoo.com/jimmy-fallon-stephen-colbert-cheer-084315450.html
0.8981	Juneteenth may finally get its due from Congress	https://news.yahoo.com/juneteenth-may-finally-due-congress-203837258.html
0.8977	Biden Signs Law Making Juneteenth a Federal Holiday	https://www.nytimes.com/2021/06/17/us/politics/juneteenth-holiday-biden.html
0.8961	These 14 House Republicans Voted Against a Juneteenth Federal Holiday	https://www.nytimes.com/2021/06/17/us/republicans-against-juneteenth.html
0.8959	Don’t Treat Juneteenth As Another Day Off. Do This Instead.	https://www.huffpost.com/entry/what-to-do-on-juneteenth_l_60c7c67be4b02df18f7f60da
0.8950	Biden makes Juneteenth a national holiday, giving federal employees Friday off	https://www.businessinsider.com/juneteenth-federal-holiday-biden-signs-bill-employees-get-friday-off-2021-6
0.8936	The Ghosts Of Comanche Crossing	https://www.texasmonthly.com/news-politics/comanche-crossing-lake-mexia-teen-drownings-juneteenth/

Table 8.7: Example 3 Similar claim from the Internet as suggestive evidence

Bibliography

H. Ahmed, I. Traore, and S. Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 127–138, Vancouver, Canada, 2017. Springer International Publishing. ISBN 978-3-319-69155-8.

H. Ahmed, I. Traore, and S. Saad. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):1–15, 2018. doi: https://doi.org/10.1002/spy2.9. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.9.

T. Alhindi, S. Petridis, and S. Muresan. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5513. URL https://www.aclweb.org/anthology/W18-5513.

P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.656. URL https://www.aclweb.org/anthology/2020.acl-main.656.

I. Augenstein, C. Lioma, D. Wang, L. Chaves Lima, C. Hansen, C. Hansen, and J. G. Simonsen. 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, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1475. URL https://www.aclweb.org/anthology/D19-1475.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, San Diego, CA, USA, 2015. URL http://arxiv.org/abs/1409.0473.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, page 107–117, NLD, 1998. Elsevier Science Publishers B. V.

K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth*

Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://www.aclweb.org/anthology/W14-4012.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. 2:265–292, 2002. ISSN 1532-4435.

L. de Alfaro, V. Polychronopoulos, and M. Shavlovsky. Reliable Aggregation of Boolean Crowdsourced Tasks. In *Conference on Human Computation and Crowdsourcing - HCOMP 2015*, pages 42–51. AAAI Press, 2015. URL https://escholarship.org/uc/item/1fz8s2tv.

G. Demartini, S. Mizzaro, and D. Spina. Human-in-the-loop Artificial Intelligence for Fighting Online Misinformation: Challenges and Opportunities. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, pages 43(3):65–74, 2020.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, MN, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

A. Fan, A. Piktus, F. Petroni, G. Wenzek, M. Saeidi, A. Vlachos, A. Bordes, and S. Riedel. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7147–7161, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.580. URL https://www.aclweb.org/anthology/2020.emnlp-main.580.

I. J. Goodfellow and O. Vinyals. Qualitatively characterizing neural network optimization problems. In *3rd International Conference on Learning Representations, ICLR*, pages 1–20, San Diego, CA, USA, 2015. URL http://arxiv.org/abs/1412.6544.

L. Graves. Anatomy of a Fact Check: Objective Practice and the Contested Epistemology of Fact Checking. *Communication, Culture and Critique*, 10(3):518–537, 2017. ISSN 1753-9129. doi: 10.1111/cccr.12163. URL https://doi.org/10.1111/cccr.12163.

M. Grootendorst. KeyBERT: Minimal keyword extraction with BERT, 2020. URL https://doi.org/10.5281/zenodo.4461265.

N. Hassan, M. Tremayne, F. Arslan, and C. Li. Comparing Automated Factual Claim Detection Against Judgments of Journalism Organizations. In *Proceedings of the 2016 Computation+Journalism Symposium*, Stanford, CA, USA, 2016. URL https://ranger.uta.edu/~cli/pubs/2016/claimbuster-cj16-hassan.pdf.

N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, and M. Tremayne. Claimbuster: The first-ever end-to-end fact-checking system. 10(12):1945–1948, 2017. ISSN 2150-8097. doi: 10.14778/3137765.3137815. URL https://doi.org/10.14778/3137765.3137815.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

A. Kazemi, Z. Li, V. Pérez-Rosas, and R. Mihalcea. Extractive and Abstractive Explanations for Fact-Checking and Evaluation of News. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 45–50, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nlp4if-1.7. URL https://www.aclweb.org/anthology/2021.nlp4if-1.7.

N. Kotonya and F. Toni. Explainable Automated Fact-Checking: A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.474. URL https://www.aclweb.org/anthology/2020.coling-main.474.

Q. Li and W. Zhou. Connecting the Dots Between Fact Verification and Fake News Detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1820–1825, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.165. URL https://www.aclweb.org/anthology/2020.coling-main.165.

C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. *CoRR*, abs/1711.05101, 2017. URL http://arxiv.org/abs/1711.05101.

R. Mihalcea and P. Tarau. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3252.

T. Mihaylova and A. F. T. Martins. Scheduled Sampling for Transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2049. URL https://www.aclweb.org/anthology/P19-2049.

S. Missaoui, M. Gutierrez-Lopez, A. MacFarlane, S. Makri, C. Porlezza, and G. Cooper. How to Blend Journalistic Expertise with Artificial Intelligence for Research and Verifying News Stories. In *CHI 2019 ACM Conference on Human Factors in Computing Systems*, pages 1–5, Glasgow, Scotland, 2019. URL https://openaccess.city.ac.uk/id/eprint/22996/.

A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, TX, USA, 2016.

Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://www.aclweb.org/anthology/D16-1244.

M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1022. URL https://www.aclweb.org/anthology/P18-1022.

A. Radford and I. Sutskever. Improving language understanding by generative pre-training. In *arxiv*, 2018.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

N. Reimers and I. Gurevych. 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, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

F. Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167, Hong Kong, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5616. URL https://www.aclweb.org/anthology/D19-5616.

N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235, 2018. URL http://arxiv.org/abs/1804.04235.

K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286, 2018. URL http://arxiv.org/abs/1809.01286.

E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro. Some Like it Hoax: Automated Fake News Detection in Social Networks. *CoRR*, abs/1704.07506, 2017. URL http://arxiv.org/abs/1704.07506.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. 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, LA, USA, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://www.aclweb.org/anthology/N18-1074.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Long Beach, CA, USA, 2017. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

W. Y. Wang. “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, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2067. URL https://www.aclweb.org/anthology/P17-2067.

Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724, 2015. URL http://arxiv.org/abs/1506.06724.