Natural Language Processing for Fact-Checking and Claim Assessment

Othman EL HOUFI
Dimitris KOTZINOS

MSc Research in Data Science & Machine Learning

March 26, 2022

Abstract

As false information and fake news are propagating throughout the internet and social networks, the need of fact-checking operations becomes necessary in order to maintain a truthful digital environment where general information can be reliably exploited whether in politics, finance or other domains. The need of this online claim assessment comes from the fact that fake news and false information can have a big negative impact on politics, economy (2016 USA Elections) and public health (COVID-19).

A number of solutions have been proposed to deal with this problem and limit the spread of false information, both manual and automatic. Of course the manual approaches done on websites such as *PolitiFact.com*, *FactCheck.org* and *Snopes.com* don't construct a viable solution for the long term as the speed and scale of information propagation increase exponentially rendering this manual fact-checking operation where human fact-checkers can't scale up at the same rate limited and incapable of solving the problem.

Here, we present our contribution in this regard: an automated solution for fact-checking using FEVER dataset as a source of truth and a state of the art language models used today for NLP tasks (BERT, RoBERTa, XLNet...) in order to classify a given claim as *Supports*, *Refutes* or *Not enough information (NEI)*. We successfully prove that fine-tuning a LM with the correct settings can achieve an accuracy of 62% and F1-score of 61% which is better than the majority of fact-checking methods that exists today.

Keywords: Natural Language Processing, Language Model, Wikipedia, Fine-tuning, Zero-shot Learning, Text processing, Natural Language Inferencing, Fact-Checking, Fake-news.

Contents

Chapter 1

Introduction

1.1 Project Context

From a social and psychological perspective, humans have been proven irrational and vulnerable when differentiating between truth and false news (typical accuracy ranges between 55% and 58%) [1], thus fake news obtain public trust relatively easier than truthful news because individuals tend to trust fake news after constant exposure (*Validity effect*), or if it confirms their pre-existing beliefs (*Confirmation bias*), or simply due to the obligation of participating socially and proving a social identity (*Peer pressure*). The social sciences are still trying to comprehend the biological motivations that makes fake news more appealing to humans.

On the other hand, the growth of social media platforms resulted in a huge acceleration of news spreading whether true or false. As of Aug. 2017, 67% [1] of Americans get their news from social media. These platforms even give the user the right to share, forward, vote and participate to online discussions. All of this made the problem of fake news spreading more and more dangerous, our economies for example, are not robust to the spread of falsity, false rumors have affected stock prices and the motivations for large-scale investments, as we witnessed after a false tweet claimed that Barack Obama was injured in an explosion which caused $130 billion drop in stock value [2]. Another recent example is related to public health where rumors about COVID-19 vaccines and drug companies influenced people in their decision on getting vaccinated.

That being said, is there a way to monitor the spread of fake news through social media? Or more specifically, how can we differentiate between fake news and truthful news, and at what level of confidence can we do that?

From a computer engineering perspective, various approaches were examined:

- **Knowledge-based Fake News Detection [3]:** a method aims to assess news authenticity by comparing the knowledge extracted from to-be verified news content with known facts, also called fact-checking.

- **Style-based Fake News Detection [4]:** focuses on the style of writing, i.e. the form of text rather than its meaning.

- **Propagation-based Fake News Detection [5]:** a principled way to characterize and understand hierarchical propagation network features. We perform a statistical comparative analysis over these features, including micro-level and macro-level, of fake news and true news.

- **Credibility-based Fake News Detection [6]:** the information about authors of news articles can indicate news credibility and help detect fake news.

In this paper we will focus on a modern approach that utilizes Language Models (LMs) for fact-checking. The goal is not to implement an algorithm that scans social networks for real time fake news detection, but rather we will design a model that can assess with a degree of confidence the truthfulness or falseness of a claim given by a user as an input by exploiting LMs that were already trained on Wikipedia, and fine-tune each LM for a downstream task in order to solve this classification problem.

1.2 Use case scenario

Suppose that while browsing the internet or talking to people you come across a claim that says *"The former U.S president John F. Kennedy died in September 22, 1963"*, as it is a general truth and not a relative truth it should be easier to verify the validity of this claim as well as find evidence that proves it.

Using the platform we will create, you can simply write the claim you like to verify with no regards to a specific linguistic rule, the model will extract relevant articles from Wikipedia using an API, then it retrieves sentences relative to your claim and apply a comparison in order to assess if the claim is True, False, or Not Enough Information as well as giving a percentage of confidence and evidence of the results that were processed straight from Wikipedia's database.

Coming back to our example, the model should return that *"John F. Kennedy died in November 22, 1963"* so the input claim is false.
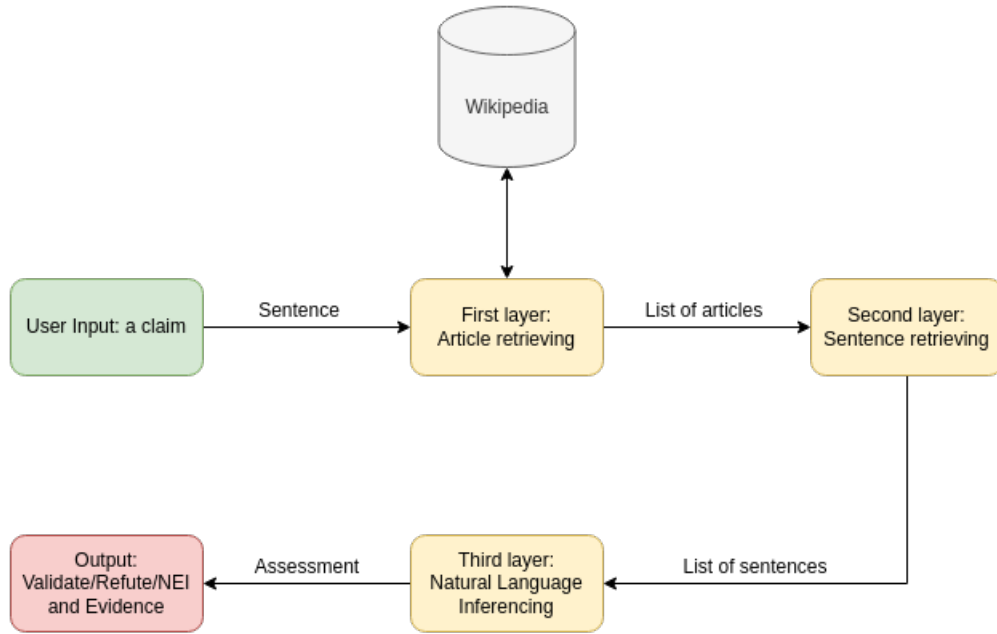

Figure 1.1: General Model Processing Pipeline

Note: this is a first draft architecture of the model to be studied, many changes will be implemented across multiple levels during the experimental procedure.

Chapter 2

Potential Datasets

The rise of fake news in the internet created the need of setting-up databases that contains as much claims as possible checked and approved manually by journalists and fact-checkers worldwide in order to be used as a dataset for experiments in automatic fact-checking models or just simply to be a reference to other journalists and the general public.

2.1 FEVER dataset [7]

FEVER (Fact Extraction and VERification) consists of 185,445 claims generated by altering sentences extracted from Wikipedia and subsequently verified without knowledge of the sentence they were derived from. The claims are classified as Supported, Refuted or NotEnoughInfo. For the first two classes, the annotators also recorded the sentence(s) forming the necessary evidence for their judgment[7].

> **Claim:** The Rodney King riots took place in the most populous county in the USA.
>
> **[wiki/Los_Angeles_Riots]**
> The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
>
> **[wiki/Los_Angeles_County]**
> Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.
>
> **Verdict:** Supported

Figure 2.1: Manually verified claim requiring evidence from multiple Wikipedia pages.

2.2 CREDBANK dataset [8]

The *CREDBANK* corpus was collected between mid October 2014 and end of February 2015. It is a collection of streaming tweets tracked over this period, topics in this tweet stream, topics classified as events or non events, events annotated with credibility ratings.the *CREDBANK* Dataset comprises more than 60M tweets grouped into 1,049 real-world events, each annotated by 30 human annotators. The label labeling of each tweet is a vector with 30 dimensions containing variable scores at five levels of veracity.

2.3 LIAR dataset [9]

LIAR is a publicly available dataset for fake news detection. A decade-long of 12.8K manually labeled short statements were collected in various contexts from *PolitiFact*, which provides detailed analysis report and links to source documents for each case. This dataset can be used for fact-checking research as well. Each statement is labeled on six levels: true, predominantly true, half-true, almost true, false, pants-fire.

2.4 FAKENEWSNET dataset [10]

FakeNewsNet is a multi-dimensional fake news data repository, which contains two comprehensive datasets that includes news content, social context, and dynamic information.

Dataset / Features	PolitiFact		GossipCop	
	Fake	Real	Fake	Real
Total news articles	432	624	6,048	16,817
News articles with text content	353	400	785	16,765
News articles with social engagements	342	314	4,298	2,902
News articles with both social engagements and news content	286	202	675	2,895
News articles with social engagement containing at least 1 reply	236	180	945	752
News articles with social engagement containing at least 1 like	283	219	2,911	845
News articles with social engagement containing at least 1 retweet	282	242	2,249	1,254
No. of tweets with replies	6,686	20,720	3,040	2,546
No. of tweets with likes	18,453	52,082	10,685	2,264
No. of tweets with retweets	13,226	42,059	7,614	5,025
Total no. of tweets	116,005	261,262	71,009	154,383

Figure 2.2: Statistics of the FakeNewsNet repository [10]

- **News content:** reliable news content retrieved from fact-checking websites and true news such as *PolitiFact* and *GossipCop*.

- **Social context:** the user engagements related to the fake and real news pieces from fact-checking websites are collected using search application programming interfaces (API) provided by social media platforms such as the Twitter's Advanced Search API.

- **Spatiotemporal information:** includes spatial and temporal information. For spatial information, we obtain the locations explicitly provided in user profiles. The temporal information indicates that we record the timestamps of user engagements, which can be used to study how fake news pieces propagate on social media, and how the topics of fake news are changing over time.

The multi-demnsionality of this dataset can be regarded as its strong value against previous datasets:

Features / Dataset	News Content		Social Context				Dynamic Information
	Linguistic	Visual	User	Post	Second order	Network	
BuzzFeedNews	✓	✗	✗	✗	✗	✗	✗
LIAR	✓	✗	✗	✗	✗	✗	✗
BS Detector	✓	✗	✗	✗	✗	✗	✗
CREDBANK	✓	✗	✓	✓	✗	✓	✗
BuzzFace	✓	✗	✗	✓	✓	✗	✗
FacebookHoax	✓	✗	✓	✓	✓	✗	✗
FakeNewsNet	✓	✓	✓	✓	✓	✓	✓

Figure 2.3: Comparison with existing fake news detection datasets [10]

2.5 Other datasets and comparison [11]

Of course other datasets exist with different structures and various sources. For each dataset we can conceptualize a new way of modeling classifiers, we can also use multiple datasets in one model.

	Content	Quantity	Labeling	Annotator
Buzzface [23]	Social media posts and comments (Facebook)	2263	Categorized on four levels (predominantly true, predominantly false, mix of true and false and no factual content)	Previously checked by news agencies (Buzzfeed)
FAKENEWSNET [15]	Whole articles	23,921	Binary (true or false)	Previously checked by news agencies (PolitiFact and GossipCop)
Fake.Br Corpus [24]	Whole articles	7200	Binary (true or false)	Considers the credibility of the source
LIAR [5]	Short statements (political)	≈12,800	Categorized on six levels (true, predominantly true, half-true, almost true, false, pants-fire)	Previously checked by news agencies (PolitiFact)
Emergent [25]	Related statements and titles	300	Binary (true or false)	Journalistic team
FEVER [26]	Short statements (Wikipedia)	≈185,000	Categorized on three levels (supported, disproved and not enough information)	Trained human annotators
CREDBANK [27]	Social network posts (Twitter)	≈60,000,000	Vector with 30 dimensions containing variable scores at five levels of veracity	Crowd-sourcing
BuzzfeedNews	Social network posts (Facebook)	2282	Categorized on four levels	Journalistic team
BuzzFeed-Webis [28]	Social network posts (Facebook)	1687	Categorized on four levels	Previously checked by news agencies (Buzzfeed)
PHEME [29]	Social media posts (Twitter)	330	Binary (true or false)	Journalistic team and crowd-sourcing

Figure 2.4: Overview of fake news datasets available [11]

Chapter 3

Related work and solutions

3.1   Language model based fake news detection [12] [13]

A paper entitled *"Language Models as Fact Checkers?"* published by a team from *FacebookAI* and *Hong Kong University of Science and Technology*, provides an example of a fact-checking model using zero-shot LM that outperforms a random baseline LM using the FEVER dataset[7].

The goal of fact-checking as mentioned previously, and relatively to this paper, is to validate the truthfulness of a given claim. Each claim is assigned to one of these labels: *Supports*, *Refutes* or *Not enough information (NEI)* to verify.

This paper describes the difference between Traditional Pipeline fact-checking models and their zero-shot fact-checking LM:

- **Traditional pipeline:** this type of models access knowledge within an external knowledge base like Wikipedia in order to validate a claim. It involves information retrieval modules such as document retrieval and sentence retrieval.

- **Zero-shot LM pipeline:** it replaces both the external knowledge base and the information retrieval modules with a pre-trained language model. In this case they used BERT as a pre-trained LM.
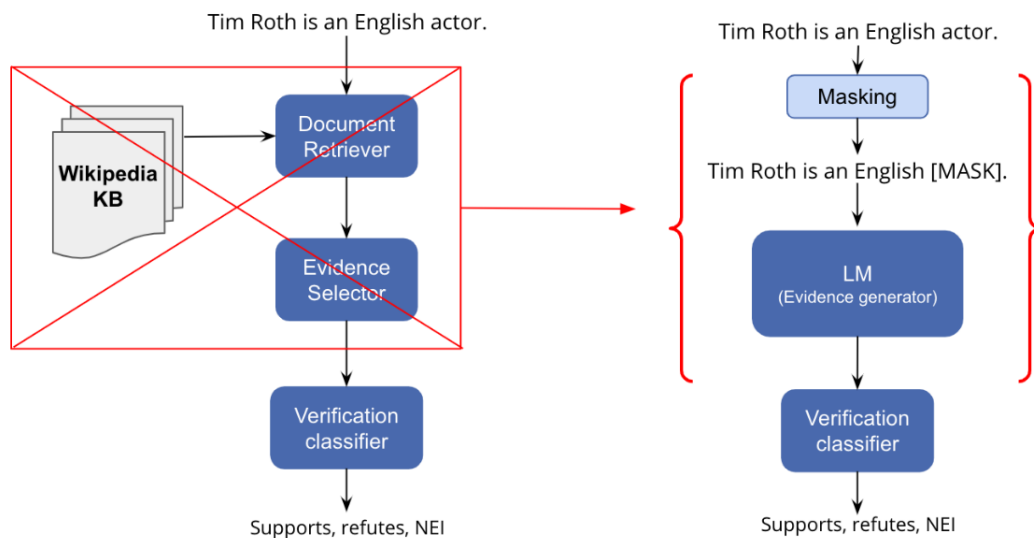

Figure 3.1: Traditional fact-checking pipeline (left) vs. zero-shot LM-based pipeline (right) [12]

To implement this end-to-end fact-checking model, they started by using an **Automatic Masking** in order to mask the last named entity in the claim. This last named entity is identified using an off-the-shelf Named-Entity-

Recognition (NER) model from spaCy. This ensures that the masked token makes use of the *knowledge* encoded in a LM, otherwise a problem like stop-words masking can occur.

Next, they used **Verification by Entailment** in order to avoid naive matching predicted an gold tokens. This method leverages a textual entailment model from Al-lenNLP to validate the LM predictions.

Claim = Tim Roth is an English actor.

Masking

Masked = Tim Roth is an English [MASK].

BERT   Predicted token = actor

Evidence = Tim Roth is an English actor.
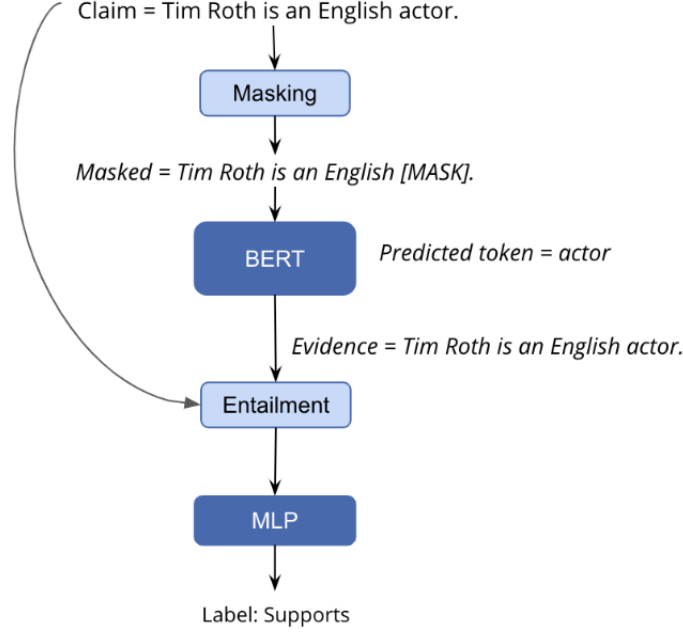
Entailment

MLP

Label: Supports

Figure 3.2: Detailed illustration of our pipeline [12]

1. Masked the last named entity found by the NER model.

2. Get the top-1 predicted token from the LM, and fill in the [MASK] accordingly to create the "evidence" sentence.

3. Using the claim and generated "evidence" sentence, obtain entailment "features" using outputs from the last layer of the pre-trained entailment model (before the softmax).

4. Input the entailment features into a multi-layer perceptron (MLP) for final fact-verification prediction.

Model	Label	prec	recall	f1	accuracy	macro prec	macro recall	macro f1
$BERT_{freeze}$	REFUTES	0.36	0.69	0.47				
	SUPPORTS	0.43	0.09	0.15	0.38	0.39	0.38	0.33
	NEI	0.39	0.35	0.37				
$BERT_{finetune}$	REFUTES	0.62	0.55	0.58				
	SUPPORTS	0.54	0.67	0.59	0.57	0.57	0.57	0.57
	NEI	0.57	0.49	0.53				
$BERTasKB$	REFUTES	0.76	0.38	0.51				
	SUPPORTS	0.41	0.92	0.57	0.49	0.59	0.49	0.44
	NEI	0.58	0.15	0.24				

Figure 3.3: Performance comparison between BERT-as-encoder models ($BERT_{freeze}$, $BERT_{finetune}$) and BERT-as-LM model ($BERTasKB$). [12]

As for evaluation, they used precision, recall, F1 and accuracy metrics in order to compare the zero-shot LM with other baseline models. They observe that the proposed approach ($BERTasKB$) outperforms $BERT_{freeze}$ on all metrics suggesting that querying language models in QA style is a better approach for extracting their encoded knowledge. Similarly, $BERTasKB$ model achieves an accuracy score of 49% which is comparable to Fever-baseline at 48.8%, except without the need for explicit document retrieval and evidence selection.

3.2 Styled based fake news detection [4]

A study done by *Piotr Przybyła* named *Capturing the Style of Fake News* in 2020 from Institute of Computer Science, Polish Academy of Sciences, in order to detect fake news or in other words assess the credibility of an article by looking at the style of writing rather than the meaning of the words and sentences.

The purpose of this study was to prove that general-purpose text classifiers, despite their good performance when evaluating simplistically, they overfit to sources of documents in training data. In contrast to this method, a truly style-based prediction that uses an analysis of the stylometric model shows that it focuses on sensational and affective vocabulary, known to be typical for fake news.

Fake news sources usually attempt to attract attention for short-time financial or political goal [14] rather than to build a long-term relationship with the reader, in this perspective, the language used by these sources tend to be informal, sensational and affective [15]. This can be used to build a classifier for indicating low credibility.

First of all they started by gathering a corpus of 103,219 documents from 223 online sources labeled by media experts like *PolitiFact* and *Pew Research Center*. Then they designed two models: a neural network and a model based on features used in stylometric analysis. This has a purpose to demonstrate that the stylometric features based model captures the affective language elements.
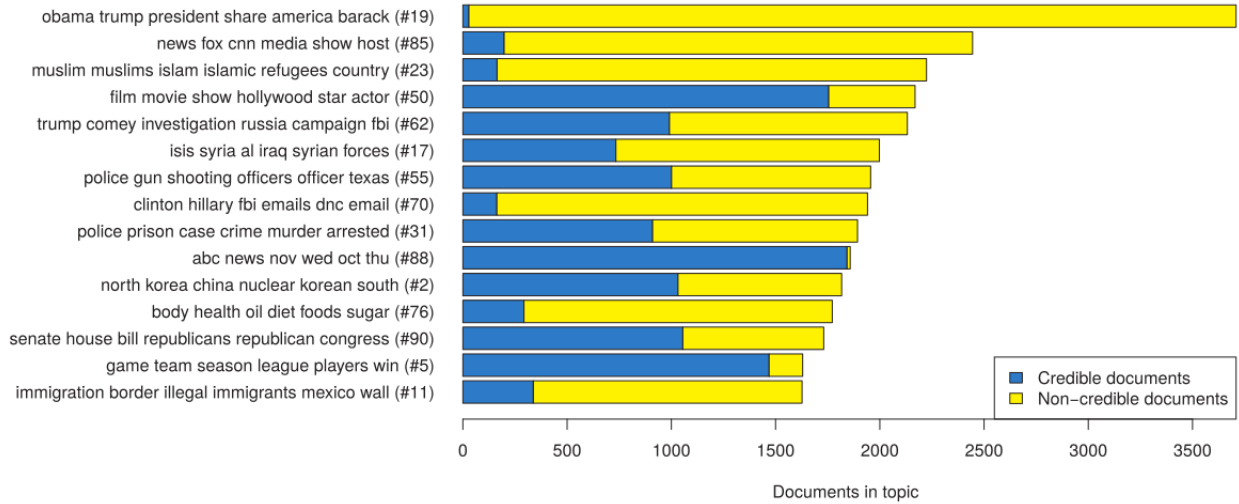

Figure 3.4: The largest 15 LDA (Latent Dirichlet Allocation) topics in the corpus, each shown with the six most significant keywords, an identifier and bars illustrating number of credible and non-credible documents associated with it [4]

Stylometric classifier: the architecture of this classifier is generally based on a collection of stylystic features followed by a linear modeling. The features are:

- number of sentences, average sentence length (in words) and average word length (in characters),

- number of words matching different letter case schemes (all lower case, all upper case, just first letter upper case, other), represented as counts normalized by the document length,

- frequencies of POS uni-grams, bi-grams and trigrams, represented as counts normalized by the document length (if present in at least 5 documents),

- frequencies of words belonging to the 182 word categories in the expanded GI dictionary, represented as counts normalized by the document length.

Neural network classifier: the applied solution was BiLSTMAvg a neural network with architecture based on elements used in natural language processing, i.e. word embeddings [16] and bidirectional LSTM (Hochreiter and Schmidhuber 1997). The following layers are included:

- An embedding layer, representing each token using a 300-dimensional word2vec vector trained on Google News,

- Two LSTM layers, forward and backward, representing each sentence by two 100-dimensional vectors (output of the last cell in a sequence),

- A densely-connected layer, reducing the dimensionality to 2 and applying softmax to compute class probability,

- An averaging layer, representing each document's class probability scores by averaging the scores for all its sentences.

The neural network is implemented and trained in TensorFlow for 10 epochs with sentence length limited to 120 tokens and document length limited to 50 sentences.

In order to understand if general-purpose text classifiers can capture document style without overfitting to features like the source or topic of document and to compare with the two classifiers they created, they evaluated two baseline models: bag of words and BERT.

The evaluation protocol consisted on running the model learning and prediction in a 5-fold cross validation (CV) scenario and comparing it's output to target labels. They used accuracy as a metric of evaluation rather than precision or recall.

Method	doc. CV	topic CV	source CV
Stylometric	0.9274	0.9173	0.8097
BiLSTMAvg	0.8994	0.8921	0.8250
Bag of words	0.9913	0.9886	0.7078
BERT	0.9976	0.9965	0.7960

Figure 3.5: Classification accuracy of our stylometric and neural classifiers compared to baselines in three evaluation scenarios, simulating, respectively, a new document from known sources and topics, a document from unknown topic and a document from unseen source. [4]

The obtained results shows that stylometric based classifier loses 10% of the accuracy on unseen sources even if it has more consistent performance over evaluation scenarios. Piotr Przybyła explains this drop in accuracy by assuming that the model specialize in the style of individual sources rather than the general style of fake news. Nevertheless, he was able to prove that his model takes into account the affective words in order to classify fake news.

3.3 Knowledge graph based fake news detection [17] [18]

Studying the *"DEAP-FAKED"* paper published in 2021 by *Mohit Mayank, Shakshi Sharma* and *Rajesh Sharma*, it turned out that by exploiting techniques related to network analysis, Natural Language Processing (NLP) and the implementation of Graph Neural Networks (GNNs) we can detect fake news with an F1 score of 88%.

The *"DEAP-FAKED"* consists of three individual components:

- **News encoder:** this component performs the contextual encoding of news title. trying both unidirectional and bidirectional sequence encoders, they implemented a 2-layer stacked biLSTM as the main subcomponent the news encoder.

- **Entity encoder:** this component identifies the named entities present in the news title and encodes the individual entities using knowledge graph (KG). For example, a news title with text *"US Officials See No Link Between Trump and Russia"*, contains two entities *"Trump"* of person type and *"Russia"* of geolocation type. The framework includes relevant entities in the news then encodes them. They used Wikidata, an open-source KG, as the source to match the entities and ComplEx KG embedding technique to embed entities.

- **Classification Layer:** this component consolidates the news encoder's and entity encoder's representations to perform the final downstream Fake News classification learning. In this framework, the two representations are concatenated to create a super representation of the news content and entities. This representation is then passed to further non-linear activated layers with decreasing layer dimensionality. The final layer consolidated the information into a single dimension which is activated by a sigmoid layer, where the final output represents the probability of the news as either true or fake.
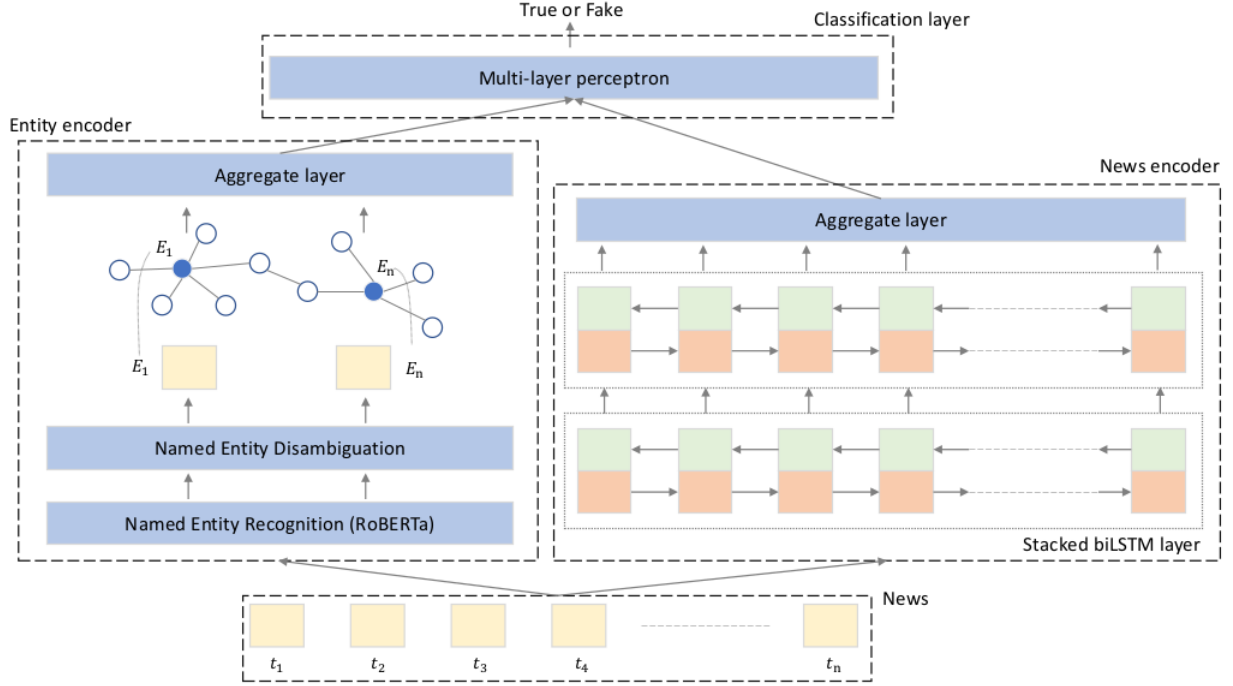

Figure 3.6: Illustration of the proposed DEAP-FAKED framework. The left sub-figure shows the entity encoder module. The right sub-figure shows the news encoder module. The top sub-figure is the Fake News classifier module. The bottom shows the tokenized news as the input to the entity encoder and news encoder. [17]

The dataset used for *DEAP-FAKED* framework can be described as following:

- **Fake News dataset:** The first dataset is the Kaggle Fake News dataset, which consists of 20,387 news items, having a near equal combination of true and Fake News. The news covers several domains such as Politics, Business, and Technology. The complete pre-processing step brought down the news item count to ∼14k with a distribution of 60% - 40% of true and Fake News classes, respectively.They called this dataset KFN-UB. The second dataset is CoAID, which contains diverse COVID-19 healthcare misinformation, including Fake News from websites and social platforms. CoAID includes 4,251 news items. After pre-processing it only 632 news item and called CoAID-UB.

- **Knowledge Graph:** the framework uses Wikidata5M as a knowledge graph, it is created by only considering the "valid" facts, where the validity is confirmed if all entities and relations in the fact have a Wikipedia article and long description.

As for the evaluation procedure, they chose five baseline models for comparison with *DEAP-FAKED* framework: ExtraTreeClassifier, LSTM, SentRoBERTa, StackedBiLSTM, and StackedBiLSTM.

Model vs Dataset	KFN-UB				CoAID-UB			
	F1 avg.	F1 std.	Acc avg.	Acc std.	F1 avg.	F1 std.	Acc avg.	Acc std.
ExtraTreeClassifier	0.7663	0.002	0.7831	0.002	0.7526	0.009	0.7638	0.008
LSTM	0.7810	0.009	0.8109	0.008	0.7255	0.021	0.7402	0.008
SentRoBERTa	0.6476	0.054	0.6879	0.040	0.7292	0.130	0.7375	0.116
StackedBiLSTM	0.7878	0.005	0.8137	0.002	0.7476	0.046	0.7585	0.030
EntWiki-StackedBiLSTM	0.8809	0.006	0.8898	0.008	0.7436	0.031	0.7454	0.032
DEAP-FAKED	**0.8866**	0.007	**0.8955**	0.007	**0.7813**	0.032	**0.7822**	0.033

Figure 3.7: Performance score of the models detailed in the paper is presented here. For each of the dataset, we report F1 macro and Accuracy metric values. We present average and standard deviation of the performance observed after performing 3 trials with different starting seed. For both the datasets, DEAP-FAKED reports the best performance value. [17]

As evident by the results, *DEAP-FAKED* reports the highest score on both of the datasets.

3.4   Hierarchical Propagation based fake news detection [5] [19]

In a paper wrote by *Kai Shu, Deepak Mahudeswaran, Suhang Wang*, and *Huan Liu* with the name of *Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation*, the detection of fake news can be done by looking at the correlation between news and the hierarchical characteristics of a social network.

The datasets used for this study were the public fake news repository *FakeNewsNet* [10] that contains data related to different fact-checking websites that offers news content, social context and dynamic information. In addition, they used the data from fact-checking websites *GossipCop* and *PolitiFact* that contains news articles with labels annotated by professional journalists. News content includes meta attributes of the news like body text, he social context includes the related user interactions with the news like posting, sharing, commenting on Twitter, and dynamic information contains the timestamps of user's interactions.

After collecting all of this data, building a hierarchical propagation network was the next step. First of all they defined two major levels of the hierarchy:

- **Micro-level:** involves users conversations towards news pieces on social media over time. It contains rich information of user opinions towards news pieces.

- **Macro-level:** encompasses information on tweets posting pattern and information sharing pattern. It represents the global propagation of information through a cascade of re-tweets.

Platform	PolitiFact	GossipCop
# True news	624	16,817
# Fake news	432	5,323
# True news with propagation network	277	6,945
# Fake news with propagation network	351	3,684
# Users	384,813	739,166
# Tweets	275,058	1,058,330
# Retweets	293,438	530,833
# Replies	125,654	232,923

(a) The statistics of FakeNewsNet
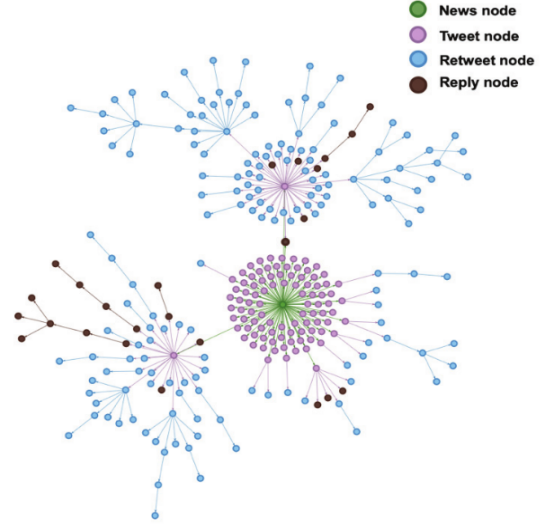
(b) An example of the hierarchical propagation network of a fake news piece fact-checked by Politifact. It consists of two types: macro-level and micro-level. The macro-level propagation network includes the news nodes, tweet nodes, and retweet nodes. The micro-level propagation network indicates the conversation tree represented by cascade of reply nodes.

Figure 3.8: From the *Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation* paper

In both these levels they used *Structural Analysis* to identify patterns in conversations, *Temporal Analysis* to understand the exchange of opinions in terms of time. For the Micro-level they added a *Linguistic Analysis* to inference the sentiment of people around news pieces, thus obtaining linguistic features for statistical comparison.

To evaluate the performance of fake news detection of the model they choose randomly 80% of news articles for training and the remaining 20% for testing. The results shows high scores for both datasets:

Datasets	Level	Acc	Prec	Rec	F1
PolitiFact	Micro	0.834	0.823	0.847	0.835
	Macro	0.807	0.816	0.789	0.802
	All	**0.843**	**0.835**	**0.851**	**0.843**
GossipCop	Micro	0.843	0.841	0.845	0.843
	Macro	0.852	0.841	0.868	0.854
	All	**0.861**	**0.854**	**0.869**	**0.862**

Figure 3.9: Evaluation of the model. [5]

3.5 Perplexity based fake news detection [20] [21]

In March 2021, *Nayeon Lee*, *Yejin Bang*, *Andrea Madotto*, *Madian Khabsa*, and *Pascale Fung* published a paper called *Towards Few-Shot Fact-Checking via Perplexity* where they propose a new approach of the powerful transfer learning ability of a language model via a perplexity score. Using a method called *few-shot learning*, they built a model that outperforms major class baseline models by more than 10% on the F1-Macro metric score.
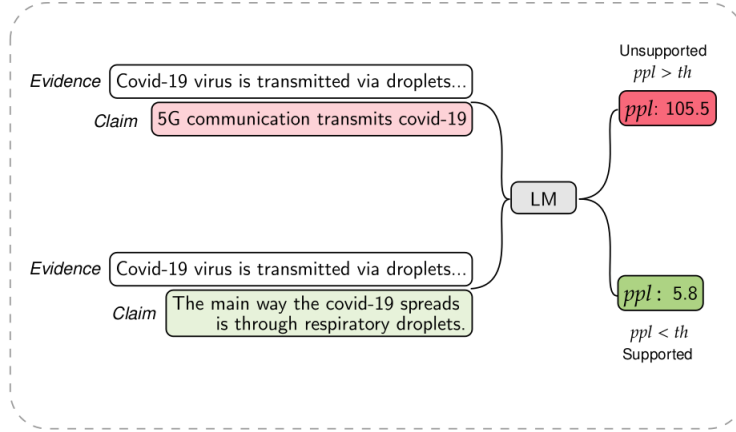
Figure 3.10: Illustration of our simple yet effective perplexity-based approach. Few-shot data samples are used to find the optimal perplexity threshold th that separates Unsupported claims from Supported claims. [20]

Few-Shot Learning refers to the technique of feeding a machine learning model with a very small amount of training data to guide its predictions, like a few examples at inference time, as opposed to standard fine-tuning techniques which require a relatively large amount of training data for the pre-trained model to adapt to the desired task with accuracy.

In NLP, *Few-Shot Learning* can be used with Large Language Models (LMs), which have learned to perform a wide number of tasks implicitly during their pre-training on large text datasets. This enables the model to generalize, that is to understand related but previously unseen tasks, with just a few examples.

Perplexity is a commonly used metric for measuring the performance of LMs. It is defined as the inverse of the probability of the test set normalized by the number of words:

$$PPL(X) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i|x_0, \ldots, x_{i-1})}}.$$

Figure 3.11: Perplexity score. [20]

Another way of interpreting perplexity is as a measure of the likelihood of a given test sentence with reference to the training corpus.

In this paper the goal is to determine the veracity of a claim given some evidence, for this they define a claim, evidence pair. The label *Supported* is assigned when relevant evidence exists that supports the claim, and *Unsupported* label for the opposite case.

Unsupported claims on average have higher perplexity than *Supported* claims. For example, *Supported* claim "Washing hands prevents the spread of diseases" has a perplexity value of 96.74, whereas the *Unsupported* claim "All dogs speak English fluently" has a much higher perplexity value of 328.23.

Unsupported Claims	Perplexity	**Supported** Claims	Perplexity
5G network can spread diseases.	826.70	Beyonce is one of the most famous singers in the world.	23.03
All dogs speak English fluently.	328.23	Chicago is one of the cities in the United States.	43.92
Washing hands **helps** the spread of diseases.	201.10	Washing hands **prevents** the spread of diseases.	96.74

Figure 3.12: Relations between veracity of claim and perplexity. Unsupported claims have higher perplexity compared to Supported claims. Note that the perplexity score listed here is using GPT2-base on each of the claims. [20]

The datasets used in this experiment are:

- **Covid19-Scientific:** a new test set constructed by collecting COVID-19-related myths and scientific truths labeled by reliable sources like MedicalNewsToday, the Centers for Disease Control and Prevention (CDC), and the World Health Organization (WHO). The set contains 172 claims with labels *(Supported, Unsupported)* obtained from the aforementioned reliable sources.

- **Covid19-Social:** a collection of 340 COVID-19-related claims fact-checked by journalists from Politifact.com. Unlike the Covid19-Scientific dataset, it contains non-scientific and socially-related claims, such as "For the coronavirus, the death rate in Texas, per capita of 29 million people, we're one of the lowest in the country".

- **FEVER:** Fact Extraction and Verification (FEVER) is a publicly released large-scale dataset generated by altering sentences extracted from Wikipedia to promote research on fact-checking systems.

For the models created for this perplexity based experiment they used one unidirectional LM and one masked LM:

- PPL_{GPT2-B} : a single-parameter classifier based on perplexity from GPT2-base [22] (unidirectional LM)

- PPL_{BERT-B} : a single-parameter classifier based on perplexity from BERT-base [23] (Masked LM)

As for baseline models they finetuned pre-trained Transformer-based [24] in order to build a classifier, which is a common approach used to achieve many state-of-the-art results in the literature.

Moving to the experimental setup, they used the *Few-Shot* technique in order to validate or train the models. Given N_D as the size of the dataset D, we do an n-shot experiment with n samples from D as a "validation set" for our perplexity-based approach or as a "training set" for the fine-tuning approach, and the remainder $(N_D - n)$ as a test set. For example in the 2-shot experiment using the Covid19-Social dataset (340 samples), we have two training samples and 338 test samples.
They took into consideration the accuracy and the F1-Macro metrics for the evaluation. Because the datasets are unbalanced, they mainly consider the F1-Macro score over accuracy as an overall evaluation.

Shot #	Models	Fine-tuning?	Size	Covid-Scientific		Covid-Social		FEVER	
				Acc	F1-Macro	Acc	F1-Macro	Acc	F1-Macro
	Major Class	N/A	N/A	58.72%	37.00%	77.35%	43.62%	50.00%	33.33%
2	BERT−B$_{ft}$	yes	110M	47.34%	32.21%	26.11%	23.33%	51.56%	37.34%
	BERT−L$_{ft}$	yes	336M	49.39%	34.80%	37.78%	27.81%	50.80%	36.49%
	RoBERTa$_{ft}$	yes	125M	52.66%	34.29%	40.75%	26.78%	50.00%	33.33%
	XLNet$_{ft}$	yes	110M	51.48%	48.49%	57.67%	44.35%	49.41%	44.65%
	PPL$_{GPT2-B}$	no	117M	**66.75%**	**64.39%**	62.61%	**53.61%**	**61.92%**	**57.50%**
	PPL$_{BERT-B}$	no	110M	47.93%	38.54%	**77.74%**	49.15%	52.54%	41.33%
10	BERT−B$_{ft}$	yes	110M	46.27%	31.70%	43.26%	30.70%	51.56%	37.34%
	BERT−L$_{ft}$	yes	336M	50.00%	36.74%	60.49%	42.18%	50.80%	36.49%
	RoBERTa$_{ft}$	yes	125M	52.64%	40.28%	40.73%	26.73%	50.00%	33.33%
	XLNet$_{ft}$	yes	110M	49.69%	42.44%	59.68%	39.45%	49.41%	44.65%
	PPL$_{GPT2-B}$	no	117M	**72.98%**	**68.57%**	**71.23%**	**55.11%**	**62.82%**	57.04%
	PPL$_{BERT-B}$	no	110M	63.15%	60.77%	61.90%	46.35%	57.59%	**57.11%**
50	BERT−B$_{ft}$	yes	110M	56.75%	53.61%	60.21%	36.91%	52.18%	38.82%
	BERT−L$_{ft}$	yes	336M	56.75%	39.15%	64.94%	44.07%	51.14%	39.99%
	RoBERTa$_{ft}$	yes	125M	56.40%	38.97%	73.13%	45.30%	50.44%	38.15%
	XLNet$_{ft}$	yes	110M	63.22%	51.98%	**77.62%**	43.70%	49.18%	48.42%
	PPL$_{GPT2-B}$	no	117M	**74.73%**	**73.83%**	73.63%	**59.91%**	**67.48%**	**64.70%**
	PPL$_{BERT-B}$	no	110M	62.53%	61.11%	71.11%	54.72%	57.44%	56.94%

Figure 3.13: Results comparison among perplexity-based classifiers and fine-tuned classifiers in 2-shot, 5-shot and 10-shot settings across three different tasks. Models whose names start with PPL are our proposed perplexity-based classifiers. Major Class is a reference to evaluate classifier performance. All test results reported are mean values of three trials with randomly selected n-shot training samples from the dataset, where $n = 2, 10, 50$. [20]

We can observe that the perplexity-based classifiers, especially PPL_{GPT2-B}, outperform all Major Class baselines across all tasks in all settings. For instance, PPL_{GPT2-B} outperforms the Major Class by a great margin of 16% and 36.8% on accuracy and F1-Macro scores. Evidence conditioned perplexity scores are capable of providing signals regarding the veracity of the the given claim.

Chapter 4

Proposed method

Most of the fact-checking algorithms today involving knowledge-based verification uses a traditional pipeline that puts in place a module for retrieving articles from an external source, another module for retrieving relevant sentences from each article and a last module for natural language inferencing (NLI) to classify a claim.
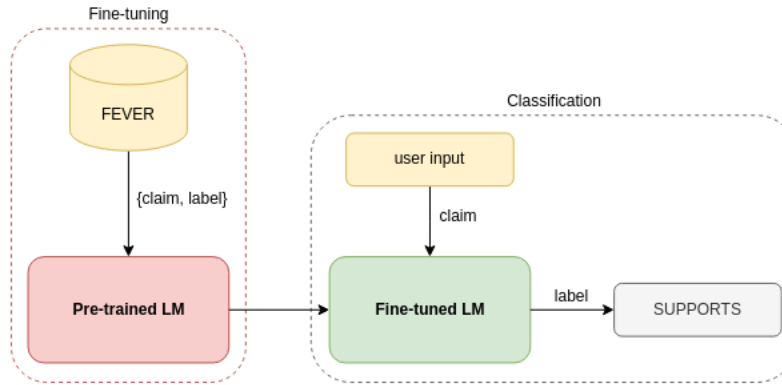

Figure 4.1: Proposed method pipeline.

In this paper we present a method that is fully reliant on the powerfulness of today's best LMs. As illustrated in Figure 4.1, we start by fine-tuning each model for the downstream task that is claim classification using the FEVER dataset, then each model is employed to assess the validity of new input claims. This approach takes into consideration only an internal knowledge source (FEVER) for fine-tuning, that is for the learning phase, which makes the prediction phase knowledge-free rather than utilizing external knowledge sources for retrieving articles and sentences.

It is also important to mention we only use LMs for classifying claims and not for generating evidence. We leave generating evidences with language models for future work.

4.1 FEVER Dataset

We will use the FEVER dataset as described before in section 2.1.

Since our mission is to classify claims to *SUPPORTS*, *REFUTES* or *NEI* and not generating evidence, we omit the evidences information in the dataset. At that point we map each label to an integer: {*SUPPORTS* : 1, *REFUTES* : 0, *NEI* : 2} and that is all we do as far as data pre-preprocessing goes.

Finally, we split the dataset to training, validation and testing sets that we can use for LM fine-tuning and testing:

ID	Claim	Label
79044	The Apple Store first opened in 2001.	1
117129	Adventure Time won an Oscar.	0
55061	Yamaha Corporation produces hardware.	2

Table 4.1: Examples of FEVER claims and labels.

Split	SUPPORTS	REFUTES	NEI	Total
Train	80,035	29,775	35,639	145,449
Val	3,333	3,333	3,333	9,999
Test	3,333	3,333	3,333	9,999

Table 4.2: Dataset split sizes for SUPPORTS, REFUTES and NOTENOUGHINFO (NEI) classes.

4.2 Language Models

The year 2018 has been an inflection point for NLP as Google introduced a LM called BERT (Bidirectional Encoder Representations from Transformers)[23]. This model was described as state-of-the-art model that solves the most difficult tasks in NLP, it is also used today in Google's search engine for text completion and translation. BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications. From there many LMs were introduced that uses the same architecture as BERT but with small changes such as number of parameters and data on which the model was pre-trained.

Comparison	BERT October 11, 2018	RoBERTa July 26, 2019	DistilBERT October 2, 2019	ALBERT September 26, 2019
Parameters	**Base:** 110M **Large:** 340M	**Base:** 125 **Large:** 355	**Base:** 66	**Base:** 12M **Large:** 18M
Layers / Hidden Dimensions / Self-Attention Heads	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 6 / 768 / 12	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16
Training Time	**Base:** 8 x V100 x 12d **Large:** 280 x V100 x 1d	1024 x V100 x 1 day (4-5x more than BERT)	**Base:** 8 x V100 x 3.5d (4 times less than BERT)	[not given] **Large:** 1.7x faster
Performance	Outperforming SOTA in Oct 2018	88.5 on GLUE	97% of BERT-base's performance on GLUE	89.4 on GLUE
Pre-Training Data	BooksCorpus + English Wikipedia = 16 GB	BERT + CCNews + OpenWebText + Stories = 160 GB	BooksCorpus + English Wikipedia = 16 GB	BooksCorpus + English Wikipedia = 16 GB
Method	Bidirectional Transformer, MLM & NSP	BERT without NSP, Using Dynamic Masking	BERT Distillation	BERT with reduced parameters & SOP (not NSP)

Figure 4.2: Comparison of BERT, RoBERTa, DistilBERT, and ALBERT. [1]

In this paper we will classify claims by fine-tuning the following LMs:

- BERT-base-uncased [23]

- RoBERTa-base [25]

- DistilBERT-base-uncased [26]

- XLNET-base-cased [27]

- ALBERT-base-v2 [28]

- BigBird-RoBERTa-base [29]

---

[1] https://humboldt-wi.github.io/blog/research/information_systems_1920/uncertainty_identification_transformers/

All of these LMs were trained on Wikipedia dataset containing cleaned articles. The datasets are built from the Wikipedia dump [2] with one split per language. Each example contains the content of one full Wikipedia article with cleaning to strip markdown and unwanted sections (references, etc.). So in one way, all these LMs were trained on facts from Wikipedia, for example if we give the following input *"Paris is the capital of [MASK]."* to BERT-base model, the output will be *"Paris is the capital of France."* with a probability of 0.951.

Therefore the conjecture behind our method is that: by fine-tuning LMs that were pre-trained on Wikipedia, and by exploiting the already stored knowledge within these LMs, we can create a self-knowledge-independent fact-checking classifier.

4.3   Experiment Protocol and Results

4.3.1   Experiment Setup

As mentioned before, we conduct our experiment on the FEVER dataset using the splits in Table 4.2. As for LMs fine-tuning we used one GPU, the NVIDIA RTX6000P-8C with 24Go of GDDR6 memory and a peak single precision floating point performance of 14,9 Tflops.

For all models we chose the following hyperparameters:

- Tokenizer max sequence length: 128

- Output layer size: 3 (classes: 0, 1 and 2)

- Activation function: GeLU

- Learning rate: 3e-5

- Optimization: Adam with linear decay

- Loss function: Cross-Entropy

- Epochs: 3

- Training batch size: 20

- Validation batch size: 20

These hyperparameters may not be optimal, but they were chosen after many different repeated experiments. Optimizing the models is left for future work.

4.3.2   Evaluation Metric

Most of the fact-checking methods that use FEVER dataset employ FEVER scoring[7], a metric that considers classification accuracy and evidence recall, but since we don't tackle the evidence problem in our approach, we rely on accuracy, recall, precision and F1-score of the model classification as metrics for evaluation.

In addition, we track other aspects of each LM such as training time, model size, and memory usage during the fine-tuning step in order to make an overall evaluation.

4.3.3   Results & Discussion

The results of the six models are reported in Table 4.3. We can observe that our approach yields better results than FacebookAI's model and most of the traditional methods that involves external information retrieval modules.

Specifically the fine-tuned *RoBERTa-base* model surpasses the fine-tuned *BERT-large* model created by FacebookAI in every metric. For instance our model achieved an accuracy score of 62% and a macro precision score of

---

[2]https://dumps.wikimedia.org/

Fine-tuned model	Label	prec	recall	f1	accuracy	macro prec	macro recall	macro f1
BERT-base-uncased	SUPPORTS	0.55	0.78	0.64				
	REFUTES	0.75	0.59	0.66	**0.62**	0.63	**0.62**	**0.61**
	NEI	0.61	0.47	0.53				
ALBERT-base-v2	SUPPORTS	0.46	0.81	0.59				
	REFUTES	0.77	0.46	0.58	0.53	0.58	0.53	0.52
	NEI	0.50	0.33	0.40				
DistilBERT-base-uncased	SUPPORTS	0.54	0.78	0.64				
	REFUTES	0.75	0.58	0.65	0.61	0.63	0.61	**0.61**
	NEI	0.60	0.47	0.53				
RoBERTa-base	SUPPORTS	0.54	0.81	0.65				
	REFUTES	0.75	0.59	0.66	**0.62**	**0.64**	**0.62**	**0.61**
	NEI	0.63	0.45	0.53				
BigBird-RoBERTa-base	SUPPORTS	0.53	0.81	0.64				
	REFUTES	0.75	0.58	0.66	0.61	**0.64**	0.61	0.60
	NEI	0.63	0.44	0.52				
XLNET-base-cased	SUPPORTS	0.53	0.81	0.64				
	REFUTES	0.74	0.59	0.65	0.61	0.63	0.61	0.60
	NEI	0.63	0.43	0.51				
Related work	**Label**	**prec**	**recall**	**f1**	**accuracy**	**macro prec**	**macro recall**	**macro f1**
BERT-large [12]	SUPPORTS	0.54	0.67	0.59				
	REFUTES	0.62	0.55	0.58	0.57	0.57	0.57	0.57
	NEI	0.57	0.49	0.53				
FEVER Baseline [30]	-	-	-	-	0.49	-	-	-
Ohio State University [30]	-	-	-	-	0.50	-	-	-
Columbia NLP [30]	-	-	-	-	0.58	-	-	-
Papelo [30]	-	-	-	-	0.61	-	-	-
UNC-NLP [30]	-	-	-	-	0.68	-	-	-
DREAM [31]	-	-	-	-	**0.77**	-	-	-

Table 4.3: Classification metrics for each fine-tuned LM using our approach vs. BERT-large fine-tuned by FacebookAI team vs. other models based on knowledge graphs and/or traditional pipelines that uses FEVER dataset (we take into consideration only the accuracy of label classification and not the FEVER scoring system).

64% which is an improvement of 5% and 7% respectively. Not only that, it is also worth mentioning that *RoBERTa-base* has only 125 million parameters and 12 encoding layers in comparison to *BERT-large* that has 340 million parameters and 24 encoding layers, thus rendering our model more efficient to train, to store and to implement.

The improvements in classification are presumably due to the fact that *RoBERTa* is pre-trained on more data than *BERT* as shown in Figure 4.2. On the other hand, our fine-tuned *BERT-base* model also achieved better results than FacebookAI's model which may be explained by the difference of hyperparameters choice.

Similarly, the fine-tuned *RoBERTa-base* model achieved more accuracy score for label classification than most of the traditional pipelines. Therefore it puts our approach on the top-10 best fact-checking models that were published by FEVER community[30] (without taking into consideration the evidence generation metric). It is also safe to say that the *DREAM* fact-checking model is certainly superior than all our LMs as it achieved an accuracy score of 77% which is 15% higher. This is a proof that there is still much room for subsequent research and improvements.

Same as FacebookAI results[12], and upon examining the results of our fine-tuned LMs closely, we also find that all our LMs struggle immensely with the NEI category (lowest F1 scores) indicating that our current approach might also need specific modules to better tackle that category.
This was also confirmed when we plotted the confusion matrix of our models (for example *RoBERTa-base* in Figure 4.3).
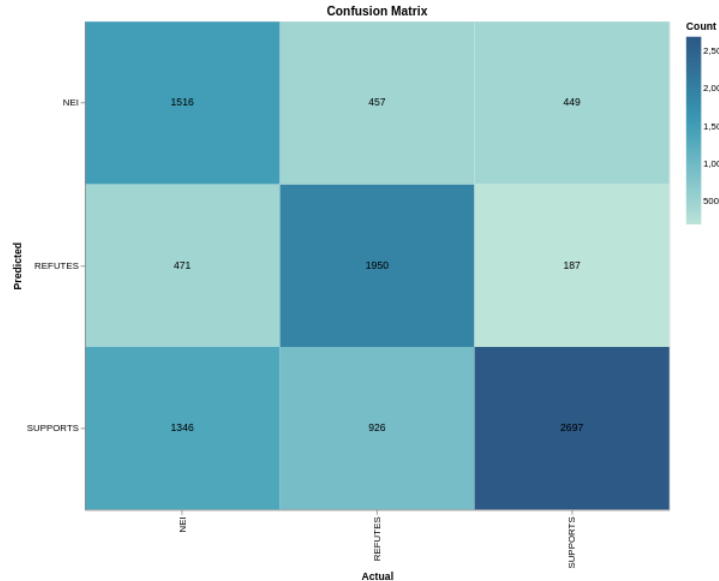

Figure 4.3: Confusion Matrix of the fine-tuned *RoBERTa-base* using test set.

Furthermore, the difference in classification performance between our LMs is not big considering the fact that each LM was pre-trained differently. The only model that performed poorly is *ALBERT-base-v2*, it achieved only 53% accuracy score and 52% macro F1 score. This can be explained by the reduced number of the model's parameters (12 million vs. *BERT*'s 110 million).

We can also see the differences between each LM during the training and the evaluation as illustrated in Figure 4.4 and Figure 4.5.
As expected, *RoBERTa-base* model reached the lowest Cross-Entropy loss of 0.878 during evaluation while *ALBERT-base-v2* sustained a higher loss of 0.961 during evaluation (also 0.75 during training). It is even more explicit in Figure 4.6 and Figure 4.7 that *ALBERT-base-v2* had the lowest Matthews correlation coefficient score (mcc) as well as accuracy score during evaluation.
Even if the other LM reached the same accuracy and mcc as *RoBERTa-base* during evaluation or a lower loss during training they did not perform as well in the test set which was demonstrated in Table 4.3.

Finally, it is also important to mention that training time differs from one LM to another (Figure 4.8) as the number of parameters are not similar, for instance *XLNET-base-cased* model spent 1h56m for training while
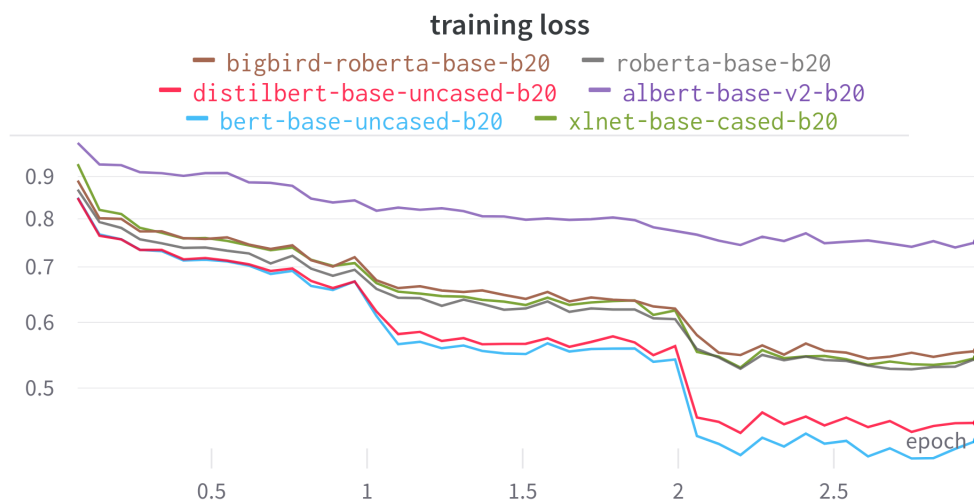
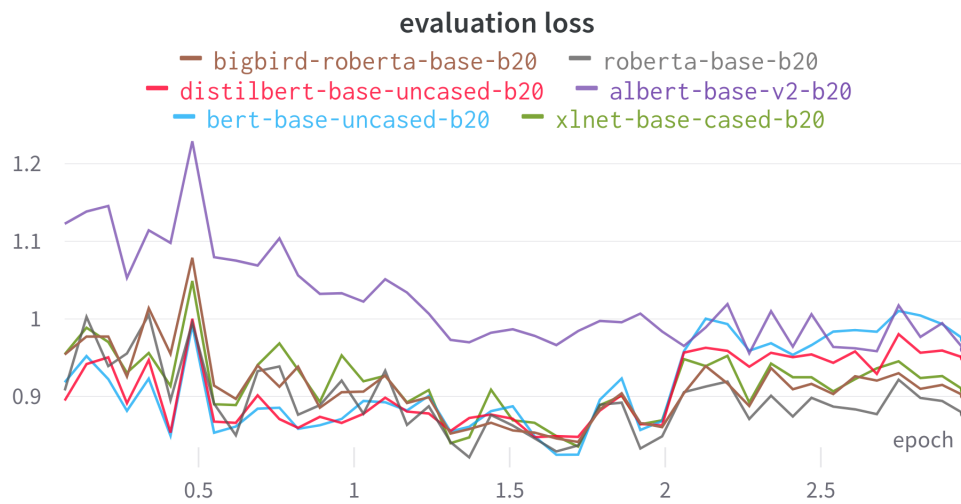Figure 4.4: The Cross-Entropy Loss of each LM during training.

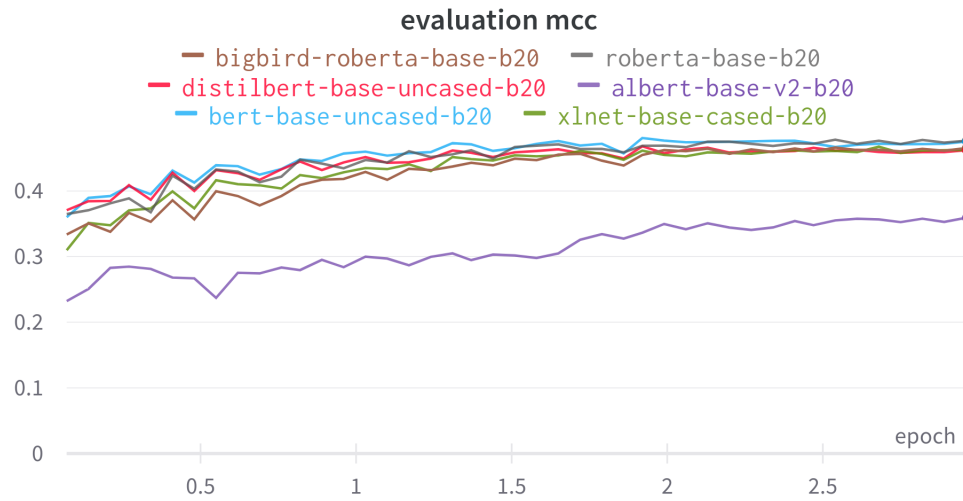Figure 4.5: The Cross-Entropy Loss of each LM during evaluation.

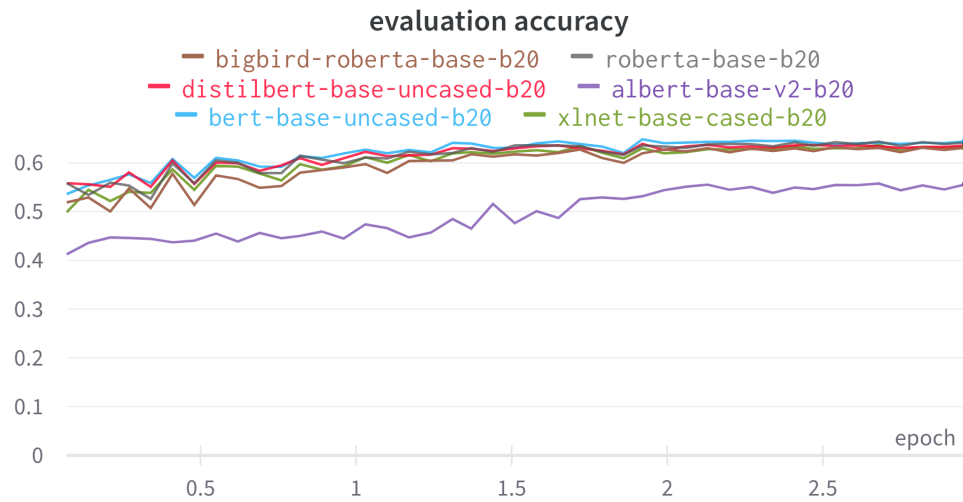Figure 4.6: The Matthews correlation coefficient score of each LM during evaluation.

Figure 4.7: The accuracy score of each LM during evaluation.

RoBERTa-base took 1h25m and achieved better results. The idea that we are trying to prove is we don't need a bigger model in order to achieve greater results for a downstream task like text classification.
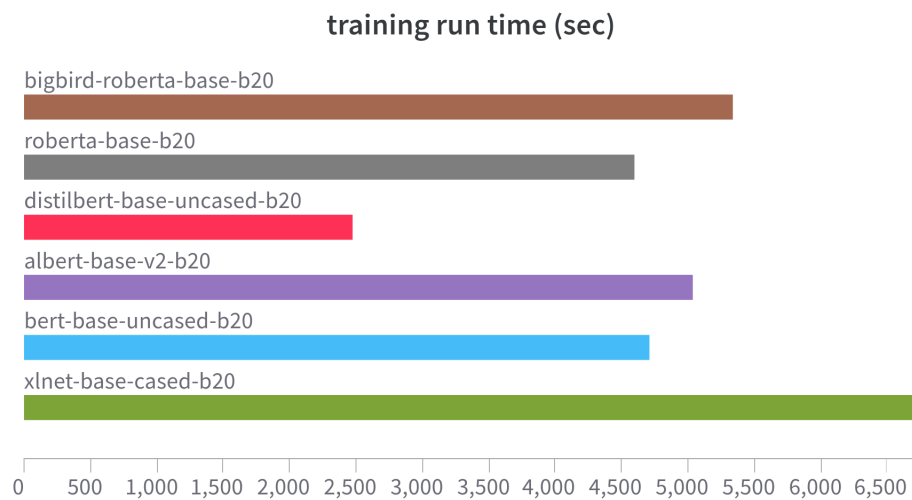
training run time (sec)

bigbird-roberta-base-b20

roberta-base-b20

distilbert-base-uncased-b20

albert-base-v2-b20

bert-base-uncased-b20

xlnet-base-cased-b20

0   500 1,000 1,500 2,000 2,500 3,000 3,500 4,000 4,500 5,000 5,500 6,000 6,500

Figure 4.8: Training time of each LM.

Chapter 5

Conclusion & Future Work

In this paper, we explored the capabilities of language models to be fine-tuned and utilized for a downstream task that is fact-checking claims. We have successfully proven the effectiveness of pre-trained language models as an independent source of knowledge rather than implementing modules for external information retrieval adopted by traditional approaches. Our experiment conclusively yields results that surpasses most of the existing fact-checking methods both traditional and LM-based. Nevertheless, our approach does not beat state-of-art traditional models leaving us with more paths to explore in order to produce a reliable fact-checking engine.

In time to come, we plan to investigate solutions to deal with NEI category where our approach struggles. In addition, we will attempt to combine other models with our approach like credibility-based or style-based models, we will also implement evidence generation alongside claim classification in order to provide the user with reliable information.

All the articles and studies exploited in this literature research are fairly recent, which proves that fake news detection or fact-checking is one of the most urgent problems to deal with today in every society. As fake news about Covid-19 vaccines rocket through the roof in social media, the World Health Organization (WHO) said in a public tweet: We're not just fighting a pandemic; we're fighting an infodemic.

Other companies like *Facebook* are *Twitter* are currently trying to make progress in fake news detection algorithms. We hope that during this project we can provide a new approach dealing with fake news propagation and contribute positively in this area.

Bibliography

[1] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 836–837, 2019.

[2] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[3] Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. Whatthewikifact: Fact-checking claims against wikipedia. *arXiv preprint arXiv:2105.00826*, 2021.

[4] Piotr Przybyla. Capturing the style of fake news. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[5] Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 626–637, 2020.

[6] Niraj Sitaula, Chilukuri K Mohan, Jennifer Grygiel, Xinyi Zhou, and Reza Zafarani. Credibility-based fake news detection. In *Disinformation, Misinformation, and Fake News in Social Media*, pages 163–182. Springer, 2020.

[7] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[8] Tanushree Mitra and Eric Gilbert. Credbank: A large-scale social media corpus with associated credibility annotations. In *Ninth international AAAI conference on web and social media*, 2015.

[9] William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[10] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

[11] Nicollas R de Oliveira, Pedro S Pisa, Martin Andreoni Lopez, Dianne Scherly V de Medeiros, and Diogo MF Mattos. Identifying fake news on social networks based on natural language processing: Trends and challenges. *Information*, 12(1):38, 2021.

[12] Nayeon Lee, Belinda Z Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. Language models as fact checkers? *arXiv preprint arXiv:2006.04102*, 2020.

[13] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[14] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[15] Vian Bakir and Andrew McStay. Fake news and the economy of emotions: Problems, causes, solutions, digital journalism pp 1-22, 2017.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] Mohit Mayank, Shakshi Sharma, and Rajesh Sharma. Deap-faked: Knowledge graph based approach for fake news detection. *arXiv preprint arXiv:2107.10648*, 2021.

[18] Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Tracy: Tracing facts over knowledge graphs and text. In *The World Wide Web Conference*, pages 3516–3520, 2019.

[19] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684, 2011.

[20] Nayeon Lee, Yejin Bang, Andrea Madotto, Madian Khabsa, and Pascale Fung. Towards few-shot fact-checking via perplexity. *arXiv preprint arXiv:2103.09535*, 2021.

[21] Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. Misinformation has high perplexity. *arXiv preprint arXiv:2006.04666*, 2020.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[27] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[28] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[29] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[30] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*, 2018.

[31] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Reasoning over semantic-level graph for fact checking. *arXiv preprint arXiv:1909.03745*, 2019.