RESEARCH PROPOSAL

A Comparative Study of Ensemble Approaches to Fact-checking for the FEVER Shared Task

Author:
Oluwabamigbe Oghenetega
ONI (691325)

Supervisor:
Prof. Terence VAN ZYL

A research proposal submitted in fulfillment of the requirements for the degree of MSc in e-Science

in the

Wits Institute of Data Science (WIDS)
School of Computer Science and Applied Mathematics

June 3, 2020

Declaration of Authorship

I, Oluwabamigbe Oghenetega ONI (691325), declare that this research proposal titled, "A Comparative Study of Ensemble Approaches to Fact-checking for the FEVER Shared Task" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this research proposal has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this research proposal is entirely my own work.
- I have acknowledged all main sources of help.
- Where the research proposal is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Abstract

Faculty of Science
School of Computer Science and Applied Mathematics

MSc in e-Science

A Comparative Study of Ensemble Approaches to Fact-checking for the FEVER Shared Task

by Oluwabamigbe Oghenetega ONI (691325)

The surge of information globally has motivated for automated rumour detection. Since misinformation is rumour on incorrect information, we use fact-checking when detecting it. The FEVER-shared task is the fact-checking task used for our comparative study. The task is divided into Document Retrieval, Sentence Selection, and Claim Verification components. We standardise TF-IDF for document retrieval, create our pipelines of one of two Sentence Selection options and one of two Claim Verification options. We then evaluate each unique pipeline on the FEVER score, compare our four pipelines to the baseline and state of the art from the FEVER Shared Task. We find that our 2-way classification task using the Siamese BiLSTM achieves better Evidence Retrieval F1 scores than the state of the art models, and that our pipeline combinations rival the state of the art for the Shared Task.

Acknowledgements

We thank our colleagues and lecturers from the University of Witwatersrand who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

We thank Brandon Sibanda for support in the implementation of the LSTM-Att Network.

I would also like to show our gratitude to the Professor Terence Van. Zyl for sharing his knowledge and adding structure to our work during the period of this study

Thanks to DST-CSIR National e-Science Postgraduate Teaching and Training Platform for their support during this research…

Contents

List of Figures

List of Tables

List of Abbreviations

CV Claim Verification. 2, 5, 9–13, 15, 20–23

ER Evidence Retrieval. 5, 8, 10–13, 15, 16, 19–23, 25

GDELT Global Database of Events Language and Tone. 2

GLOVE Global Vectors of Word Representations. 7

LSTM Long Short-Term Memory Networks. 6, 9

MLP Multi-Layer Perceptron. 9

NER Named Entity Resolution. 12, 22

NLP Natural Language Processing. 6

PCA Principal Component Analysis. 14, 19

RNN Recurrent Neural Networks. iv, 5, 6

RST Rhetorical Structure Theory. 9

RTE Recognizing Textual Entailment. 8, 9, 16, 19

TF-IDF Term Frequency - Inverse Document Frequency. iv, 7, 8, 13, 16, 22

VSM Vector Space Models. 7, 9

Word2Vec Principal Component Analysis. 7, 13, 14

Chapter 1

Introduction

The exponential surge of information on the internet has brought many solutions to humanity but has magnified other problems faced by society at large. One such problem is the increased spread of rumours. Verifying rumour in its various terms and concepts has always been important, however, the need has been increasingly apparent since U.S President, Donald Trump, called Fake News to well established news sources, like CNN, during the electoral campaign in 2016 [1]. Statistics show that 67% of Americans use social networks as their main platform for news. Hence, the thought of tens of millions of social network users reacting to false rumours with likes, shares and comments is not foreign [1]. For this reason, the World Economic Forum warned that one of society and democracy's largest growing threats will be Fake News [2].

1.1 Growth of Fake News

In 2016 & 2017, Fake News was awarded "word of the year" by Oxford Dictionary and Collins Dictionary, due to its enormous impact on politics and society in general. Social networking sites like Facebook, Twitter and the likes, that encourage free speech and sharing across global communities prey on human cognition and behaviour by amplifying biases and behaviours of individuals by its community's bias, therefore creating an effect of an Eco-chamber for emotions, content propagation and in essence Fake News propagation. Fake News thrive in these platforms through repeated exposure causing confirmation bias. Since Fake News is mostly biased news written with malicious intentions to manipulate reader behaviour, it succeeds in its exploitation [3][1].

Some of these effects can be seen in May 2017, as Qatar's state news agency's hacked Twitter account released series of comments allegedly by the Emir, causing neighbouring countries to sever diplomatic ties with it [2]. In 2013, a depletion of $130 billion in stock value was caused by claims stating that Barack Obama was injured in an explosion [1]. In other scenes, a rumour published by BuzzFeed stating that a Jewel store in the U.S replaced real diamonds with fake ones caused a 3.7% drop in the Jewel stores' brand stock [2].

1.2 Types of Fake News

According to Zhou *et al.* (2018), a broad definition of *Fake News is False news* [1]. A narrow definition of *Fake News is intentionally and veritably false news published by a news outlet*. Although the term Fake News has recently gained popularity, its many forms are well known. Each one with slight nuance distinguish them from one another. According to Zhou *et al.* (2018), the different types of unverified information and claims can be distinguished by three main categories: (*i*) Authenticity (correctness), (*ii*) Intention (emotionally manipulative), and (*iii*) News (newly received information) [1]. As shown in Table 1.1, different combinations of these categories yields different types of claim and in essence different solutions for them. In literature, the terms Fact-checking, rumour detection, click-bait detection, Claim Verification (CV) and Fake News detection are used interchangeably to describe Fake News detection.

TABLE 1.1: Types of Fake News [1]

	Authenticity	**Intention**	**News?**
Maliciously False news	False	Bad	Yes
False news	False	Unknown	Yes
Satire news	Unknown	Not bad	Yes
Disinformation	False	Bad	Unknown
Misinformation	False	Unknown	Unknown
Rumour	Unknown	Unknown	Unknown

For example, disinformation is false information [news or non-news] with a bad intention aiming to mislead the public.

1.3 Public Attempts to Fake News Detection

Over the past few years, different organizations and government have increased focus on Fake News detection. Many websites, such as PolitiFact, Channel4 and Snopes make available their fake news dataset labelled by editors for public use [4]. Other strides include the launch of The FakeNewsChallenge to acquire submissions that could use stance detection to classify Fake News appropriately. Browser Plugins (NewsScan) are now available to verify news as a reader reads on a browser. MediaBiasFactCheck.com is a website that provides a "bias score" for a news article. This is also used in the NewsScan plugin. The GDELT project (Global Database of Events Language and Tone) is one that pulls in broadcast news in 100+ languages to support streaming of data. FakeNewsDetector.org uses their robot, Robinho, to detect and flag Fake News, click baits and extremely biased news by linking directly to twitter and facebook feeds through an installed browser extension [5][6]. More detailed approaches to the variants of Fake News detection will be discussed in later sections.

However, the portion of Fake News Detection labelled Fact-checking is our area of interest. Journalism defines Fact-checking as the task of assessing the truthfulness of a claim made in a written or spoken language. Thorne *et al.* (2018) further simplifies Fact-checking into three points as a task that "addresses a claims logic, coherence and context" [7].

The FEVER dataset was chosen to support our study. The dataset contains 185,455 claims acquired from Wikipedia by editing (falsifying, rewriting, extracting) sentences and then subsequently verifying them using Wiki pages. The claims were labelled based on one of the three classes SUPPORTED, REFUTED, or NOT ENOUGH INFO. Along with the respective labels are the lists of ids that cross reference with Wikipedia pages id from the Wiki pages dataset. Wiki pages contains extracted introductory section of the Wikipedia articles. The data collection was done by Thorne *et al.* (2018) in two tasks called Claim Generation and Claim Labelling [8]. The tasks were given to a group of 50 annotators. A 5-way annotator agreement was done by randomly selecting four percent of claims, which were not skipped, to be annotated again. The Fleiss Kappa score was 0.6842. The precision and recall obtained on the dataset for Evidence Retrieval was 95.42% and 72.36% respectively. The claims dataset was then split as shown in Table 3.1.

The problem definition of the FEVER Shared Task also suits the Fact-checking problem we aspire towards as all evidences for a claim is required for its correct classification. The rationale behind this decision is explained further in the proposal. Our stance is that appropriate evidence extraction is imperative to Fact-checking on misinformative claims.

Hence, in our research we highlight current approaches to Rumour detection, Fact-checking and Fake news detection. We consider dataset options, and explain why we chose FEVER Shared Task dataset, how our problem relates, and the significance of the research. We present and discuss our methodology, evaluated pipelines and results. Finally we conclude with stand out remarks and future work.

Chapter 2

Background and Related Work

2.1 Background

Human cognition and behavioural theories engineered in economics, psychology, philosophy and social sciences provide qualitative and quantitative metrics we can use to study reactions to different Fake News forms. The types of Fake News evident in Table 1.1 is known as rumour. According to the definition specified in Table 1.1, rumour is an unverified claim with an unknown intent. Due to its nature, rumours have the property of being partly true as opposed to just true and false, hence for appropriate classification it is essential to understand what type of rumour needs to be detected to increase the probability of successfully detecting it using these behavioural theories [9].

2.1.1 Perspectives to Rumour Detection

Success in the different perspectives (based on Human Cognition and Behavioural theories) to rumour analysis in literature are dependent on the type of rumour to be detected. Their justification and explainability is based on the theories acquired from these disciplines. We summarise rumour analysis as follows: style-based Rumour Analysis (analyses how rumour is written); network-propagation based Rumour Analysis (analyses how rumour spreads); account-based Rumour Analysis (analyses account roles in rumour dissemination around a subject) and knowledge-based Rumour Analysis (analysis that focuses on false knowledge in Fake News) [1][5]. Knowledge-based approaches are related to Fact-checking. The main aim of this perspective is to assess news authenticity by justifying or negating it with previously verified content. This category works well on Misinformation and False news and is considerably harder to perform. We will delve more into Fact-checking in subsequent sections.

To appropriately classify rumours, one needs to ensure that the correct labels are given for the tasks. The varying format of rumour, mainly due to partially true information, makes it very difficult to model as a binary classification problem. Oshikawa *et al.* (2020) reviewed a number of Fake News corpi, where each one had a varying number of labels for the classification task. The challenges with labelling this dataset occurs during aggregation of editor or journalist labels. Due to varying ideas of what is true and not true. The complexity also increases with the length of the claims [4]. Logical approaches to test aggregation correctness and annotator agreements include Fleiss Kappa (k) score for a labelled corpi [8].

2.1.2 Fact-checking on Misinformation

We separate Fact-checking from all other types of Fake News detection since it is verification of misinformation. CV is one of the task required to perform Fact-checking [7]. The other task is Fact-Extraction, which involves extracting required factual evidence that are used for verification [1].

In the FEVER Shared Task paper, Thorne *et al.* (2018) defined three task for Fact-checking. These are Document Retrieval, Sentence Selection and CV. Thorne *et al.* (2018) specified Document Retrieval as the process of collecting k most similar documents to a claim. Document Retrieval involves selecting the most relevant sentences required to support that claim, while CV involves using the extracted sentences to decide the verdict on the specified claim. The tasks Document Retrieval and Sentence Selection are encapsulated in Fact-extraction. Due to the varying meaning of Fact-extraction we will use the term Evidence Retrieval (ER) to denote this process [8].

For Fact-checking systems to be usable and effective these systems are required to be: real-time; accurate; interpretable; simple; scalable; and should evaluate based on the larger context [10].

2.1.3 Algorithmic concepts applied to Fact-checking

To enable Fact-checking we require algorithms whose underlying principles allow for ER and CV. Below are a few algorithmic concepts we plan on exploring in our study. We will describe in detail the intuition behind them, and will then review their applications in literature, along with other Fact-checking attempts.

Recurrent Neural Networks (RNN) variants

RNN are feedforward neural networks with recurrent edges (edges that span adjacent time steps) [11]. In these networks, the output $\hat{y}^{(t)}$ of a given node $x^{(t)}$ is affected by recurrent edges with $x^{(t)}$ and $h^{(t-1)}$ hidden node values, if it has one, as well as $h^{(t)}$. Therefore input $x^{(t-1)}$ can affect output $\hat{y}^{(t)}$ [11]. Figure 2.1 depicts a simple RNN with recurrent edges across time steps.

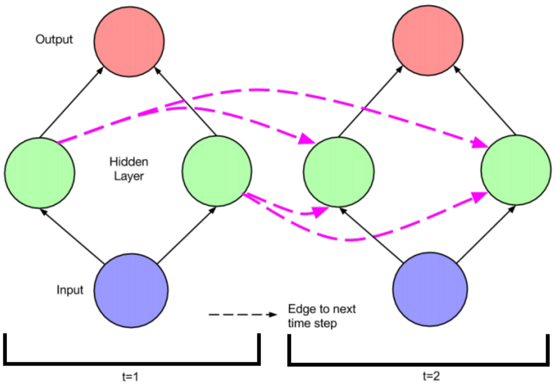

FIGURE 2.1: Unfolded RNN [11]

Due to its structure, the main benefit of an RNN is that previous time steps have a more pronounced effect on the current time step. However, as researchers tried to obtain better performance with the RNN, the vanishing gradient problem was discovered [11]. Many variants were tried to solve this problem including the introduction of a Rectified Linear Unit (RELU), that does a $max(0, x)$ on node value. One of the most successful RNN variants for solving the vanishing gradient problem was the Long Short-Term Memory Networks (LSTM) [9]. The LSTM has been used severally for word-sequence learning problems and has achieved better success at these tasks. A variant of the LSTM called the Bidirectional-LSTM has also been successful at this tasks. Its rationale is based on using future and past information to influence outputs.

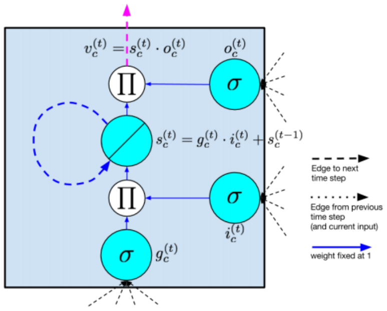
$v_c^{(t)} = s_c^{(t)} \cdot o_c^{(t)}$
$o_c^{(t)}$
Π
σ
$s_c^{(t)} = g_c^{(t)} \cdot i_c^{(t)} + s_c^{(t-1)}$
Π
σ
$i_c^{(t)}$
σ
$g_c^{(t)}$
Edge to next time step
Edge from previous time step (and current input)
weight fixed at 1

FIGURE 2.2: One LSTM memory cell. The self-connected node is the internal state s. The diagonal line indicates that it is linear, i.e. the identity link function is applied. The blue dashed line is the recurrent edge, which has fixed unit weight. Nodes marked $\prod$ output the product of their inputs. All edges into and from $\prod$ nodes also have fixed unit weight.
[11]

LSTM Architectures vary due to their input node - a standard input node with a recurrent edge where summed weights are computed through an activation function, input gate - a sigmoidal unit that takes activation product from $x^{(t)}$ and $h^{(t-1)}$ allowing flow of values from the input node that are from zero to one memory cell components in the hidden layers of the network, an internal state - the memory cell with linear activation as a result of its recurrent edge with fixed unit weight (solving the vanishing gradient problem), the forget gate - flushing the contents of the internal states, and the output gate value - the product of the internal state value by the output gate [11]. This architecture is depicted in Figure 2.2. LSTM-Att (Attention Network) are a variant of LSTMs with an Attention Layer. They produce a weight vector with a product of word-level and sentence level vectors from state t [11].

Siamese Neural Networks

Every type of neural network can be adapted into a siamese architecture making it a Siamese network [12]. A siamese architecture is made up of the same network copied and merged with an energy function. An example of an energy function is the triplet loss function. The data input of a siamese network should be in the form (x_1, x_2, y). In the case of natural modelling x_1 and x_2 are similar while y $\in$(0,1) where $y = 0$ is dissimilar and $y = 1$ is similar. When training we intend to minimise the distance between similar pairs and maximise the distance between dissimilar pairs.

When applied to Natural Language Processing (NLP), Siamese RNNs or Siamese LSTMs have been successful due to their persistence capability (ability to retain context) [12]. They also require lots of data to generalise well.

Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF is a well-known metric for ranking documents based on word importance [8][13]. Given a query, TF-IDF calculates word importance based on word frequency in a document and offsets the word importance score by the frequency of words in the corpus the document is in [14]. Word importance is calculated per word, therefore for a given query (sentence) various ranking methods can be used to decide on document importance to a given sentence. The closer the value is to one the higher its importance. The TF-IDF Equation (2.3) is as follows for term(t):

$$TF(t) = \frac{d(t)}{d} \tag{2.1}$$

$$IDF(t) = log_e(\frac{D}{D(t)}) \tag{2.2}$$

$$TF - IDF(t) = TF(t) \times IDF(t) \tag{2.3}$$

The count of term t in a document will be denoted as $d(t)$, the number of terms in a document (d), the total number of documents (D), and Number of documents with term t in it ($D(t)$). This technique is usually applied to Document Retrieval and has the advantage of partial matching of a query to a document. Its disadvantages includes its inability to capture semantics in sequences. This means sentences like a document saying *"Adam hates Eve"* would mean the same as *"Eve hates Adam"*. TF-IDF is considered a Vector Space Model (VSM). VSM is the generic class used for all models that involve the representation of documents as a vector of the terms represented in the document [15].

Words as Vector Embeddings

One can express words as vectors using embeddings such as using GLOVE and Word2Vec. They both obtain vector representations of words by obtaining co-occurrence stats of words from a corpus. The main difference is that GLOVE is unsupervised using Nearest Neighbour calculations through cosine similarity or euclidean distance to acquire word embeddings, as opposed to Word2Vec which uses a semi-supervised method to create vector representations of words. The training process aims to discriminate target words from noise words, this is usually done using logistic regression [16][17].

GLOVE calculates the ratio of probabilities so that the ratio of non-discriminative words reduces and that of discriminative words increases [16]. Essentially it aims to learn word vectors so that their dot product is the logarithm of the word's probabilities of occurrence. Word2Vec can be approached as a Continuous Bag of Words model or a Skip-Gram model. The Continuous Bag of Words model predicts a word w_t from a context h_t, meanwhile the Skip-Gram model predicts h_t given w_t. It uses maximum likelihood estimation to estimate these probabilities [17].

$$J_{NEG} = logQ_\theta(D = 1|w_t, h) + k \underset{\tilde{w}_t \; P_noise}{\mathbb{E}}[logQ_\theta(D = 0|\tilde{w}_t, h)] \tag{2.4}$$

Word2Vec tries to maximise the objective given in Equation (2.4), where the $logQ_\theta(D = 1|w_t, h)$ is the binary logistic regression probability of h occurring given term w_t in dataset D. From the noise distribution (p_{noise}) we draw k contrast words.

Unlike TF-IDF they both help us acquire semantic representations of words therefore making synonyms and antonyms acquisition possible.

TABLE 2.1: Fact-checking dataset options. FTR-18 has tweets (2,064k) and news articles (3,045)

Name	Text length	Size	Labels	Properties
LIAR [9]	short	12,836	six	No evidence, but contains justification
FEVER [19]	short	185,445	three	All evidence and label given per claim
FNC [5]	short - medium	28,866	four	Contains stances and labels. Possibly be phrased as evidence
FTR-18 [20]	short - medium	3,045 + 2,064K	three	Contains stances and labels. Possibly be phrased as evidence

Public Datasets for Fact-checking

A number of Fact-checking datasets are available for rumour detection research. However most of them are focused on news and therefore Fake News Detection. Some datasets highlighted by Oshikawa *et al.* (2020) and Papadopoulou *et al.* (2017) were for general Fake News detection without need for Evidence Retrieval [4][18]. We noticed that these five data sets contained enough information to be phrased as a Fact-checking problem. In Table 2.1 are the datasets, their properties and reasons for selection.

Although the FakeNewsChallenge (FNC) dataset and the FootballTransferRumours-18 (FTR-18) are good options for our study as they accommodate stance detection and veracity classification, we however chose the FEVER dataset because our problem is an ER and Veracity Classification problem.

2.1.4 Related Work on Fact-checking

Several approaches to Fact-checking have been taken, both inline with the FEVER Shared Task challenge and outside of it. Ranging from feature extraction of sentences to deep learning. We will highlight some of these below.

FEVER Baseline Model

Thorne *et al.* (2018) created a baseline model for the FEVER Shared Task that uses the Document Retrieval and Question Answering system derived from TF-IDF vectors with binned unigram and bigrams and applied cosine similarity before ranking sentences by their similarities and then performing RTE on the sentences using an MLP [8].

Features extracted in literature

Naderi *et al.* (2018) reported the use of feature based models using parts of speech, bag-of-words, entity types, LDA topics, sentiment and many more in Fact-checking models. However, it is worth noting that a feature-based model containing metadata tends to exploit small biases in the data collection of rumours. These features are usually fed into ML models for further analysis [4][13].

Classification using deep learning

Oshiwaka *et al.* (2020) reports rhetorical approaches to Fact-checking using RST on VSM models with the main idea being to find the center of true and false news in high-dimension RST space. They also report different ML approaches (supervised and unsupervised) to this classification task. Naive Bayes classifier, Support Vector Machine, Logistic Regression and Random Forests are some ML algorithms that have some promise through their proficiency at the classification task. Convolutional Neural Networks have also been seen to work with graph-like data for the task. A common occurrence is the use of linguistic word count features added to pre-trained word embedding like Word2Vec and GloVe as inputs into these neural networks. It has been reported that these methods acquire higher accuracy than the Naive Bayes classifier [4][5].

Highlights of Submissions to the FEVER competition

Thorne *et al.* (2018) reported results and submissions to the FEVER Shared Task competition [18]. The highest ranked team UNC-NLP acquired a FEVER score of 64.21%. Thorne *et al.* (2018) reported details of their pipeline [18]. They found that Majority of the teams participating did not deviate from the baseline model pipeline of Document Retrieval, Sentence Selection and Natural Language inference. For the initial search most teams looked towards Named Entity, noun phrases and capitalised expressions on the corpus. However, the top team used page viewership to exploit bias in the dataset construction.

TF-IDF and string matching using named entity matching was seen as the best Document Retrieval method. The three main approaches to Sentence Selection were keyword matching, supervised classification and sentence similarity scores. For Supervised classifications an LSTM was used by Team Athene [21]. They modified the Enhanced Sequential Inference Model LSTM (LSTM-ESIM) as specified in Hanselowski *et al.* (2018) [21]. Evidence combination was also done by concatenation in some cases and one approach used a MLP for the combination. For the RTE component, some of the algorithms used are Random Forests, LSTM-ESIM, Decomposable Attention, Transformer Model [22] and feature extraction methods using non-lexical features [18].

CV is usually structured as a supervised learning problem and not an unsupervised problem. We also use this approach in our methodology.

2.1.5 Problem Definition based on FEVER

Thorne *et al.* (2018) [8] specified the high-level problem definition as verification of textual claims against textual sources. When this task is compared to the problem statement, its main difference is in the fact that questions contain the information required to provide the answer (i.e. find the evidence). However, statements are more generic and more work is needed to collect evidence to support or refute it. The FEVER dataset, therefore, contains claims that need to be classified against Wikipedia pages as SUPPORTED, REFUTED and NOT ENOUGH INFO. It is expected that systems return the evidence supporting or refuting a claim, however this is not required for the NOT ENOUGH INFO class. The FEVER Shared Task is interested in the accuracy of verification on one-hand, but also (mainly) interested in the correctness of the evidence retrieved.

2.2 Problem Statement

For interpretability [4], accuracy [10], simplicity [10], scalability [10], and facilitation of new reading comprehension methods [4], it is important that Fact-checking algorithms extract evidence related to a claim and then reason based on the context of its extracted evidence to classify a claim. However, poor accuracy have been acquired with current Fact-checking algorithms when the requirements for ER are: $k = 5$ documents, and $l = 5$ sentences, while requirements for successful CV is evidence necessity.

2.3 Significance and Motivation

We will be evaluating key approaches to ER and CV, and comparing the pipelines formed from their combinations to the winning submissions for the FEVER shared tasks. We will find out which approaches perform significantly better than deep learning approaches to ER, and we will observe how selected evidence affects the classification of our varying CV tasks.

2.4 Research Aims and Objectives

2.4.1 Aims

Due to the overall poor performance of the state of the art algorithms on the FEVER dataset for CV using evidence as of 2019. The aim of this study is: (i) to investigate to what extent ER (Document Retrieval and Sentence Selection) affects the performance of CV algorithms on FEVER by comparing the accuracy obtained from pairs of ER and CV algorithms, (ii) to observe if there is a statistically significant improvement, in classification accuracy, obtained from the best pair formed from the combination of ER and CV algorithms.

2.4.2 Objectives

To achieve the aim specified, the two tasks involved: ER and CV will be approached as follows:

1. Use TF-IDF for document similarity (Document Retrieval) to extract five nearest pages to a claim;
2. Implement ER algorithm identified: Five nearest pages will be used as input to Sentence Selection algorithms (Siamese/BiLSTM Network and BiLSTM-Att (RTE)), where word embeddings will be based on Word2Vec;
3. Evaluate accuracy of Evidence Retrieval algorithms with reference to Fully supported accuracy in a similar manner to Thorne *et al.* (2018) [8];
4. Implement CV algorithms identified. CV Algorithms: Random Forest (feature extraction), Recognizing Textual Entailment (BiLSTM-ESIM);
5. Evaluate the accuracy of Claim Verification algorithms concerning NearestP for ScoreEv (Retrieved Evidence Score Evaluation) accuracy in a similar manner to Thorne *et al.* (2018) [8].

2.5 Research Questions

1. Amongst the selected ER method, which method extracts the most relevant evidence from Wiki pages, where $1 <= numberofevidence <= L$ and relevance is calculated with the percentage of fully supported documents?

2. How do Fact-checking models (Pair of ER and CV) with the best ER component perform on ScoreEv accuracy, F1 score, precision and recall, for SUPPORTED, REFUTED or NOT ENOUGH INFO classification types, with NOT ENOUGH INFO Sentence Selection (NearestP) in comparison to other Fact-checking models developed in this study?

2.6 Delineations, Limitations and Assumptions

2.6.1 Based on FEVER

For appropriate comparisons across this study, the baseline model and winning submissions, the study will be subject to all delineations, limitations and assumptions imposed by FEVER.

Hence, evaluation of algorithms will be done in line with Thorne *et al.* (2018) [8][23]. Although the Kappa Fleiss value of the dataset is 0.68, we consider all labels assigned to a claim and the evidence given as the only evidence available in the corpus to verify a given claim. Note that due to the current mean length of 9.4 tokens per claim, we understand that the results observed might not be re-achievable in a claim corpus with a significantly larger length. We will also perform some recommendations outlined by research, for example, not adding already verified knowledge to the corpus as this might bias results.

2.6.2 Based on Research Process

For our research process several assumptions and scope changes were made. We assume that the FEVER dataset was collected with care and that the dataset online currently reflects the state specified in Thorne *et al.* (2018) [8]. We will limit the scope to considering Wikipedia articles' introductory pages that are currently in the FEVER dataset and no other source will be used for evidence. ER and CV algorithms specified will be implemented or replicated according to what was done in their relevant literature, only hyper-parameter tuning might differ.

Chapter 3

Research Methodology

3.1 Introduction

3.1.1 Research Design

We performed a comparative study between combinations of ER and CV algorithms to observe to what extent CV is affected by ER, and to see if there was a statistically significant improvement in accuracy acquired from the best ER and CV pair. The algorithms cover a set of approaches spanning from deep learning to machine learning. Our comparison due to a set of standardised processing done on the claims and Wiki corpus brings novelty.

3.2 Methodology

3.2.1 Research Instruments

The FEVER Shared task pipeline was created in Python. We chose Python because of its exhaustive deep learning libraries, and open source nature. Word2Vec has pretrained word embeddings on the Wikipedia corpus available. TensorFlow and Keras deep learning frameworks have Python APIs and libraries making it easier and viable to develop these models and compute them on scalable on-demand infrastructure. The following libraries and models were readily available: TF-IDF, Word2Vec, and Spacy.

Other deep learning models that were used are Siamese BiLSTM Network based with a ReLu activation function [12], a bidirectional LSTM model with an attention network [21], Semantic Role labelling algorithm, Named Entity Recognition NER [24], Random Forests, and Bidirectional LSTM with Enhanced Sequential Inference Models [12]. All deep learning architectures were replicated using tensorflow, keras and pytorch as a framework.

The FEVER dataset was already curated and tested for completeness for the FEVER AI Shared Task. The data from the source was already split into train, validate, test. However, we performed audit checks on the data to ensure that the number of entries in the different datasets corresponded to the numbers specified in literature [8].

We compare across our ER models, CV models and their combination pairs to the baseline model and top three team's models. Their scores are shown in Table 4.10. We compared along precision, recall, F1 scores, label accuracy and FEVER scores.

The FEVER data set was acquired as specified by Thorne *et al.* (2018) [8].

TABLE 3.1: Dataset split sizes for SUPPORTED, REFUTES and NOTENOUGHINFO classes [8]

Split	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,639
Dev	3,333	3,333	3,333
Test	3,333	3,333	3,333
Reserved	6,666	6,666	6,666

3.2.2 Data

One advantage we have is that the input data is already split as shown in Table 3.1. We process the Wiki pages to acquire relevant documents before acquiring the relevant evidences to every single claim. As a result, our pipeline classifies a claim as SUPPORTED, REFUTED and NOT ENOUGH INFO and provides all evidence used to deliberate on the claim.

3.2.3 Analysis

As highlighted in the aims of this study, our two main goals are to (i) compare across ER algorithms and find the ER algorithm that extracts the most relevant evidences (sentences) from the Wiki pages corpus, and (ii) evaluate the pairs obtained from ER and CV combinations inline with the benchmark models. To effectively perform ER we needed to do Document Retrieval and Sentence Selection. Given claims we perform textual pre-processing on the text field by removing any special characters, and other forms of text normalization. Once the text is prepared for Document Retrieval we extract all meaningful words from the text and pass the text as parameters into our TF-IDF to acquire a score for each document and then retrieve the claim's most related documents.

After pre-processing on the entire dataset. We continue the following phases with our training dataset. Note that to find the best generalisation we performed grid search hyper-parameter tuning on all trained models. We settled on hyper-parameters that generalise well on the validation set in terms of precision and recall. Our Methodology is depicted in Figure A.1.

Methodology and Evaluation for Evidence Retrieval

In order to increase the Document Retrieval rate of our TF-IDF implementation, we performed textual cleaning and manipulation that improved team submission's Document Retrieval. We added the title of every document to its text. We then used Spacy to identify Named Entities in claims, capital expressions were also identified. Each sentence term's score per document was calculated and then aggregated. The five highest scoring documents that contained the identified entities by sum of TF-IDF score for a claim were selected for the next steps. Amazon RedShift was used for this computation.

The selected documents was broken up into sentences based on the appearance of newline characters. The Word2Vec 100 dim english vector embeddings were used to encode the text as they were fed into our architectures. Option one's problem was rephrased as a binary classification problem.

Siamese Bidirectional LSTM

For our binary classification problem we define similarity based on evidence. Given a claim and evidence pair ($< Cl, Em >$), if Em is used as evidence (SUPPORTED, REFUTED) to Cl and another claim and evidence pair ($< Cm, Eo >$) is used as evidence, then the Siamese LSTM considers them as similar, else they are dissimilar. We concatenated the claim and evidence and passed it to the Word2Vec embedding layer using the Siamese Bidirectional LSTM for training. The $l = 5$ most similar sentences are selected. As depicted in ER Algo one in Figure A.1.

LSTM Attention Networks

For option two we use LSTM-Att network to recognise textual entailment and train the network on the three way classifications provided in the training datasets. We consider the evidence with the highest RTE probability's verdict, if its verdict is SUPPORTED or REFUTED we extract only the top five verdict of the same type or NOT ENOUGH INFO. This gives us coherent evidences i.e. they are all (SUPPORTED or NOT ENOUGH INFO) or (REFUTED or NOT ENOUGH INFO) or (NOT ENOUGH INFO but not SUPPORTED or REFUTED). As depicted in ER Algo one in Figure A.1.

We then go ahead and evaluate accuracy of ER algorithms with reference to Fully supported document evaluation in a similar manner to Thorne *et al.* (2018) [25].

Methodology for Claim Classification

We concatenate retrieved evidences and a claim as one sentence and pass them into the Claim Verification algorithms. For our three way classification tasks, we consider two Claim Verification algorithms, namely random forests [4] with feature extractions [18] , Recognizing Textual Entailment using a LSTM-ESIM in line with Hanselowski *et al.* (2018) [21].

Random Forests
Since Random Forests have been found to work well for classification tasks, given the claim and evidence set pair, we will extract features that are grammatical and morphological (as opposed to stylistic features used for style-based rumour detection). Some of the features we will use include number of characters, number of words, average word length, pronouns, common words, PCA POS Histogram features, percentage of stop words and many more, we use the select K-best features to select the best combination of features to acquire the best classification accuracy.

LSTM with Enhanced Sequential Inference Modelling
For the RTE model our implementation will be identical to Hanselowski *et al.* (2018) implementation. The LSTM-ESIM uses attention and pooling operations as outlined [21].

Training of all these models was done on sets of 145,449 claims with their distribution across the classes as specified in Table 3.1. Hyper-parameter tuning of algorithms was performed on the validation set.

The recorded accuracy of CV algorithms in the next sections will be with reference to NearestP for ScoreEv (Retrieved Evidence Score Evaluation) accuracy in a similar manner to Thorne *et al.* (2018) [8]. With this method, every SUPPORTED or REFUTED classification is required to have the evidence required for its verdict, if not the classification is incorrect. For the class labelled NOTENOUGHINFO, its classification is independent of the evidence retrieved.

All Evaluation will be done on the test set.

We will report, discuss, compare and evaluate accuracy, recall and F1 scores for statistically significant improvements to the benchmarks highlighted in literature [11][8][21].

3.3 Conclusion

In conclusion, we develop six unique pipelines for Fact-checking with combinations of ER and CV algorithms as specified above. We will evaluate to what extent ER affects a CV algorithm performance. We will also compare across pipelines and the benchmarks to see if a statistically significant improvement in accuracy, recall, F1 score is obtained by any of our six pipelines. We then discuss alternative recommendations, steps forward, and future work in the literature.

Chapter 4

Results

4.1 Evidence Retrieval

The two components of ER are Document Retrieval and Sentence Selection. Across the FEVER Shared task most teams stuck to the baseline model's pipeline structure of Document Retrieval, Sentence Selection and RTE. The joint process of Document Retrieval and Sentence Selection are referred to as Evidence Retrieval. The results highlighted below were calculated based on our algorithms performances on the paper's claim test set of 9999 claims (3333 each of SUPPORTED, REFUTED and NOT ENOUGH INFO). Hence, they are comparable to model performances highlighted in the FEVER Shared task [23].

4.1.1 Document Retrieval

For this phase, we only considered $K = 5$ documents to be retrieved. We chose TF-IDF as our standard for Document Retrieval, since Team Papelo reported the highest precision (92.18%) and F1 score (64.85%). We repeated the claim transformation steps that gave them the most significant improvement, as specified in Malon *et al.* (2019) [26], but were unable to replicate their score of 81.2% for Document Retrieval. Our Document Retrieval score was 49%.

We could not replicate Team Papelo's results due to our approach to TF-IDF calculations. We considered a Named Entity in a text (claim or Wiki-page) as one word rather than its word composition. We also used sum as the aggregate for our TF-IDF sentence scores. Team Papelo did not state their method of aggregation.

Using Spacy and capitalised expressions to filter our results set we improved the Document Retrieval rate in comparison to the standard TF-IDF module by 20%, and then found a significant improvement of 5% after adding the document title to the text. We compared between totaling, maximums and averaging the scores of each word in a document. TF-IDF totals performed best with estimated 5% more on Document Retrieval. Hence our 49% accuracy as shown in Table 4.1.

TABLE 4.1: Document Retrieval Changes

Changes	**Effect**
Standard TF-IDF Module	24%
TF-IDF + Spacy + Cap Exp	44%
TF-IDF + Spacy + Cap Exp + Title	49%

We calculated TF-IDF scores for Named Entities, with more than one word, as one word.

4.1.2 Sentence Selection

In this phase, we also extract $l = 5$ most relevant sentences from each document retrieved for a claim from the Wikipedia dump. We however consider two different classification tasks. The first performs a 2-way classification with the labels $VERIFIABLE$ (if the sentence is in the evidence set of a claim), and $NON-VERIFIABLE$ (if the sentence is not in the evidence set of a claim). The second considered a 3-way classification task to the Evidence Retrieval based on the claim's label, in terms of Evidence Retrieval it only selects a group of coherent evidences per claim as mentioned in the methodology section.

During text preparation, we removed words from texts that denoted punctuation, for example -LRB- was used in the Wikipedia dump for '('.

Siamese Bidirectional LSTM

The Siamese bidirectional LSTM aims to pick out similarities between claim-evidence pairs. For this we used a Rectified Linear Unit as the activation function rather than Triplet loss or sigmoid function, increasing the convergence rate and a testing accuracy to 77.15%, based on the claim and evidence pairs ability to enable a $VERIFIABLE$ or $NON-VERIFIABLE$ verdict. The training vs validation losses displayed in Figure 4.1 shows the model adjusts to fit the training set, meanwhile it declines steadily in validation error as epochs increase. The algorithm converges as the number of epochs approach 30 epochs.

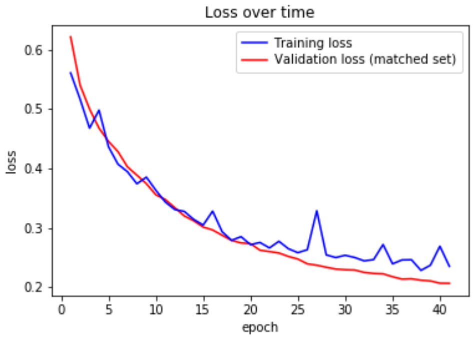

FIGURE 4.1: Siamese BiLSTM training loss vs. validation loss over Epochs

A sequence length of 50 words for our bidirectional LSTM was selected because the text field to be processed was the claim and evidence pair. The average length of a claim was 9.4 and the 75th percentile of the evidences was 53. Hence we selected 50 as the max sequence length, to capture as necessary information as possible from the acquired text.

For the 2-way classification task our precision, recall and F1 scores are 71.08%, 80.6%, 75.5% respectively in terms of fully supported documents (i.e. evidence retrieved). The ROC AUC score was 0.775.

TABLE 4.2: Siamese Bi-LSTM Confusion Matrix

	True	**False**
VERIFIABLE	83724	28776
NON-VERIFIABLE	**21744**	90756

This model better identifies when evidences are insufficient to verify a given claim as shown in Table 4.2 .

Bidirectional LSTM Attention Networks

The Decomposable Attention LSTM was found to be one of the stand out performers for recall and so was selected as an option for Sentence Selection. For the 3-way classification task we set up the architecture by adding an embedding layer with a maximum length of 70 words, due to concatenated claim-evidence pairs. We add an Attention decoder layer with 50 units and an output vocab size of 250, before adding a BiLSTM of five units, and finally the SoftMax three unit dense layer to converge using categorical cross-entropy. We see in Figure 4.2 that as the training loss reduces, the validation loss slowly plateaus, as both lines cross at epoch five. We performed early stopping at eight epochs, due to the large deviation between the training and validation loss.

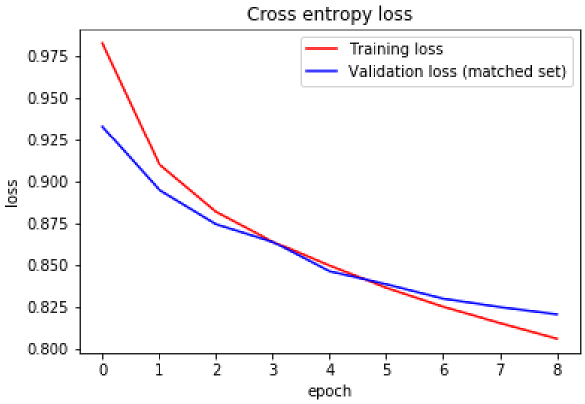

FIGURE 4.2: BiLSTM Att training validation loss over Epochs

As mentioned previously we acquire the top coherent evidences and then calculate our scores. For the 3-way classification task, we achieved an accuracy of 44.6% on testing. Our precision, recall and F1 scores are 46.93%, 42.33%, 44.67%, respectively. This was similar to the results received by Hanselowski *et al.* (2018) [21]. When evidences are not coherently consolidated our precision, recall and F1 scores are 42.82%, 49.60%, 46.21% respectively.

We notice that this penalised recall of 44% was as a result of the complexity involved in the 3-way classification task. The model struggled to appropriately classify the NEI and REF classes. Overall the model had poorer recall. Recall is important because it evaluates how much evidence we can identify.

TABLE 4.3: BiLSTM-ATT Confusion Matrix

	Asg. SUP	Asg. NEI	Asg. REF
True SUP	1805	527	1001
True NEI	2539	278	516
True REF	1664	943	726

This model struggles with recall as most verdicts are assigned to the NOT ENOUGH INFO class as shown in Table 4.3.

4.2 Claim Verification

For Claim Verification, a 3-way classification task is performed by both algorithms. We did not consider unsupervised learning approaches for this task as it was a NLI/RTE problem, but also to standardise with the majority approach from the FEVER Shared Task.

In order to get to one final classification for a claim, we concatenate claim and sentences in one text and our ER takes this as input. The UNC-NLP team acquired the best results using this approach, it also provides one verdict from the evidences [27]. Our sentence representations are in vector embeddings and extracted feature sets as is necessary for our chosen algorithms. The LSTM-ESIM was said to perform best on this task.

For the training dataset we used our ground truth dataset and used NearestP to acquire documents for claims labelled NOT ENOUGH INFO.

4.2.1 Random Forests

We extracted features from the claim-evidences text, based on [18], that indicate the verdict relationship between the claim and the evidences. For this we consider 27 features that were cut down to 17. After using PCA on the POS Histogram features. We used the select K-best features function and acquired 12 features. These per sentence were char count, word count, uppercase char count, mean word length, common words count, ratio of stop words to texts and PCA1 to PCA6 extracted from the POS Histogram.

For the RTE task our accuracy on the test set accuracy was 33.32% with precision, recall and F1 scores at 33.95%, 33.00% and 33.32% respectively. The confusion matrix is tabulated in Table 4.4.

Although the errors acquired improved when the Siamese BiLSTM was used as the Evidence Retrieval component, we suffered poor performance with the NearestP dataset used. This was due to the grammatical and morphological features used in this study. These features were previously used for the Click Bait study and we expected it to be adaptable to misinformation, unfortunately these expectations did not follow through [18]. Extracting more relevant misinformation features should improve our model accuracy. This indicates that the combination of evidences provided was better suited to the features used. These results are discussed further in the Section 5.

TABLE 4.4: Random Forest Confusion Matrix

	Asg. SUP	**Asg. NEI**	**Asg. REF**
True SUP	1213	1179	941
True NEI	323	1158	1852
True REF	1225	1179	930

4.2.2 LSTM Enhanced Sequential Inference Model

For the Bidirectional LSTM-ESIM model we used Word2Vec embeddings, and set the maximum sequence length to 150. We then used a drop out rate of 0.5 and learning rate of 0.0004. The hyperbolic *tan* function was selected as the activation function for the 3-way classification task.

TABLE 4.5: BiLSTM-ESIM Confusion Matrix

	Asg. SUP	Asg. NEI	Asg. REF
True SUP	804	992	1537
True NEI	149	1047	2137
True REF	1714	695	924

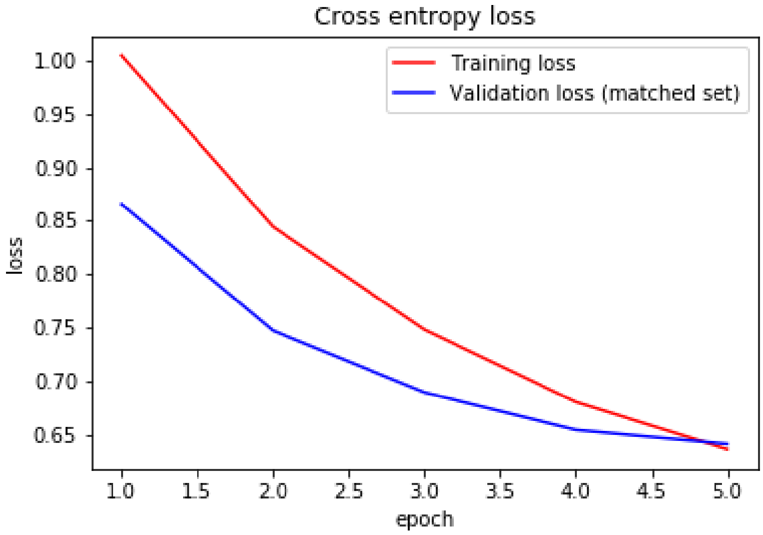

FIGURE 4.3: BiLSTM-ESIM training loss vs. Validation loss over Epochs

Hence the model's test accuracy was 57.60%, the training and validation loss is as depicted in Figure 4.3. The precision, recall and F1 scores are 58.4%, 57.6%, 57.02% respectively.

4.2.3 Study FEVER Pipelines

Although, the previous model outperforms our Random Forest alternative. We were also interested in how inputs from one model could significantly impact the output of another. For this reason, we created four different pipelines using the combinations of the ER and CV algorithms made available. The confusion matrices for the BiLSTM-ATT + BiLSTM-ESIM, SIAMESE-BiLSTM + BiLSTM-ESIM, BiLSTM-ATT + RF, SIAMESE-BiLSTM + RF are tabulated in Tables 4.6, 4.7, 4.8 and 4.9 respectively.

Our consolidated results are depicted in Table 4.10. All Neural network models were trained and scored on GeForce GTX 1060 6GB/PCIe/SSE2 graphics cards.

TABLE 4.6: BiLSTM-ATT + BiLSTM-ESIM Confusion Matrix

	Asg. SUP	Asg. NEI	Asg. REF
True SUP	2287	278	768
True NEI	1876	1120	337
True REF	3029	91	213

Table 4.7: Siamese-BiLSTM + BiLSTM-ESIM Confusion Matrix

	Asg. SUP	**Asg. NEI**	**Asg. REF**
True SUP	726	1689	919
True NEI	1817	750	766
True REF	307	587	2438

Table 4.8: BiLSTM-ATT + Random Forest Matrix Confusion Matrix

	Asg. SUP	**Asg. NEI**	**Asg. REF**
True SUP	698	1005	1630
True NEI	149	1047	2137
True REF	804	992	1537

Table 4.9: Siamese BiLSTM + Random Forest Matrix Confusion Matrix

	Asg. SUP	**Asg. NEI**	**Asg. REF**
True SUP	1975	409	949
True NEI	782	981	1570
True REF	1997	455	881

Table 4.10: Pipelines results, models developed are marked ×

Model Name	**Evidence Retrieval (%)**			**Label**	**FEVER**
	Precision	**Recall**	**F1**	**Accuracy (%)**	**Score (%)**
UNC-NLP	42.27	70.91	52.96	**68.21**	**64.21**
Athene UKP TU	23.61	**85.19**	36.97	65.46	61.58
Siamese + BiLSTM-ESIM ×	71.08	80.6	**75.5**	57.6	59.4
Papelo	**9f2.18**	50.02	64.85	61.08	57.36
Siamese + RF ×	71.08	80.6	**75.5**	33.32	39.30
BiLSTM-ATT + RF ×	46.9	44.7	42.3	33.32	32.81
FEVER Baseline	11.28	47.87	18.26	48.84	27.45
BiLSTM-ATT + BiLSTM-ESIM ×	46.9	44.7	42.3	57.6	23.7

4.2.4 How Results Align with Research Questions

We observed that the best ER algorithm is one that extracts relevant classification evidence from a text. A 2-way classification task is preferred due to its ability to simplify the problem space, as opposed to distinguishing between types of evidences. Hence, we achieved an F1 score of 75.5%, improving on the results acquired from the state of the art Evidence Retrieval models.

We also measured the extent to which ER affects CV. We first ensured consistency with text pre-processing making results comparable, between ER - CV pairs. We achieved superior gains in FEVER scores between pipelines where the Siamese BiLSTM was used as opposed to the BiLSTM-ATT for Evidence Retrieval. This results support the notion that the best Fact-checker contains the best individual ER - CV pipeline. We also find that evidence recall is critical for an accurate Fact-Checker, this is perhaps due to the fully supported document evaluation case (required evidence for appropriate classification).

Chapter 5

Discussion

The FEVER Shared Task was conducted by Cambridge research. The necessary due diligence was performed during data extraction, collection and annotation to ensure that the curated dataset was fit for purpose i.e. fact extraction and verification of claims.

Due to the poor performance of state of the art models on FEVER, our main aim is to investigate to what extent ER affects the performance of CV. In order words, does collecting appropriate evidence improve our pipelines ability to make a verdict?

In Thorne *et al.* (2018) [23] concluded that it was difficult to assess the impact of submitted models due to differences in the text preprocessing phases. However, after standardising preprocessing with our implementations we can compare, evaluate and discuss model performance as follows.

For our Document Retrieval component we were only able to retrieve 49% of the documents required to verify a claim as opposed to the 81.2% reported, this was possibly due to our identification of Named entities in the text and performing TF-IDF on the entity as opposed to the words in an entity's name [26]. Another factor that could have influenced this was the absence of a retrieved document with the keyword "film" along with the top 5 documents. In our case, the sum was used to aggregate the TF-IDF scores per term. We found that using the sum of TF-IDF scores worked well as opposed to the max and mean. Team Papelo did not mention their method of aggregation.

However, we found that named entity extraction contributed to our model accuracy by roughly 20%. We used Spacy's NER model, noun phrasings and capitalised expression, hand in hand for the NER component. Our capitalised expression components extracted entities in the instance that it was not picked up by Spacy. We acquired 49% accuracy while attempting to extract evidences for SUPPORTED and REFUTED claims, as a result the NOT ENOUGH INFO class did not require evidence extraction. Its exclusion could have caused the percentage of change in accuracy. However, due to our inability to replicate the results, we provided appropriately sampled and aggregated data to our ER models during training. The set of evidences used also followed the NearestP methodology. For ER we only considered fully supported documents as evidence retrieved.

The BiLSTM-Att Model as our first ER model was a 3-way classification task, we adequately trained our model as shown in Figure 4.2. We stopped the training process after eight epochs, as we noticed a lot of overfitting happens past eight epochs. Due to its classification task and its requirement for coherent evidences, we see an increase in accuracy by two percent when coherent evidence is required. Its confusion matrix shows consistent misclassification of the NEI and REF classes into SUP this leads to a penalised recall of 44%. The least understood class by this model is the NEI class.

The Siamese BiLSTM was a 2-way classification task that was trained for 40 epochs as shown in Figure 4.1. The 77.1% accuracy acquired supports the notion of similar evidence regardless of its verdict being SUP or REF. The confusion matrix in Table 4.2 shows that the network can appropriately distinguish between the 2 classes with better accuracy with the NON-VERIFIABLE class. This is perhaps due to VERIFIABLE information and its potential to be broken up into SUP and REF.

Comparing across our ER algorithms we obtained the best accuracy from the Siamese Bi-directional LSTM with 77.15%, this is possibly because its 2-way classification task only specifies if a given sentence is an evidence or not, greatly simplifying the problem space. However, the LSTM-Att is a 3-way classification task that requires the presences of only high scoring coherent evidences to be retrieved. Hence, when comparing across the two ER models we expect that the Siamese BiLSTM would have a higher recall (80) than the LSTM-Att (49.6). We expected the LSTM-Att to have a better precision score than the Siamese Bi-LSTM due to its 3-way classification task, which was false.

In comparison to the state of the art and baseline metrics, the Siamese BiLSTM acquired good recall but the LSTM-Att did not outperform the Baseline model in recall. Recall is important because it evaluates how much evidence we can identify. The Siamese BiLSTM F1 score (75.5) outperforms the state of the art Papelo (64.85) - TF-IDF for Document Retrieval and string matching using named entities and capitalized expressions. Hence, portraying the potentials of the 2-way task for Evidence Retrieval.

The ROC AUC of the Siamese BiLSTM is 0.775. This means that the Siamese BiLSTM 77.5% of the time, is capable of distinguishing between those sentences that are evidences to a claim and those that are not.

The Claim Verification algorithms use the evidences retrieved as inputs during the verdict process. Essentially we consider label accuracy when checking the performance of the model on a dataset of ground truth, and FEVER score when considering the input from the preceding ER component.

The Random Forest CV algorithm being a machine learning algorithm is slightly more robust as a result of its engineered features. It achieved a label accuracy of 33.32% which means an evenly split odd across the three classes. Perhaps better feature extraction and the Bag of Words model could better improve its scoring. The Random Forest test accuracy improved when given data from the Siamese BiLSTM network. This indicates that the combination of evidences provided was better suited to the features used.

We performed early stopping on the BiLSTM-ESIM model at five epochs. We achieved the highest label accuracy of our study of 57.60%. The results show that often SUP and NEI classes are misclassified as REF with this model. LSTM-ESIM variants were used by the top 3 teams in their implementation of the RTE component. The LSTM-ESIM is the state of the art of RTE, and in this case our implementation provided better overall label accuracy, with precision, recall and F1 scores all above 57%. The main reason of its high performance nature is its encoding and alignments layer that computes on score for each elements in two vectors before a matching is performed using neural networks and then propagated to the max-pooled output layer, therefore increasing its ability to retain context and enhance inference.

Since all components in our pipeline are setup, we calculated our FEVER scores and compared it to the benchmarks and state of the art. We noticed that models that have a high FEVER score usually required a high recall for their Evidence Retrieval component. This supports the notion that models find it harder to predict RTE when more relevant evidences are available. The highest scoring FEVER Score is still the submission of the UNC-NLP group. We observed that the Random Forest was less rigid on its classification tasks in comparison to the ESIM. As depicted in Table 4.10, the Siamese BiLSTM + BiLSTM-ESIM outperformed team Papelo on the FEVER Score, however the BiLSTM-ATT + BiLSTM-ESIM performed even worse than the baseline on the FEVER Score, perhaps due to bad recall.

Random Forest results however remain consistent. This is because the relevant features extracted from text like Parts of Speech and word counts are not as influenced by context, in comparison to the BiLSTM-ESIM network that checks for alignment within a text.

Although our end to end pipelines did not outperform the FEVER Score, however there exist the potentials for the individual pipeline components in further work, improving on the benchmark and state of the art.

Chapter 6

Conclusion

In conclusion, we discuss how Fact-checking is a type of content-based rumour detection relating to a portion of an overall fake news detection problem [5]. We highlight many issues caused by the spread of fake news and why it has become of increased importance to automatically detect it. We consider the FEVER Shared Task, highlight the problem background it attempts to solve, and evaluate the benchmark, state of the art models, and proposed pipelines.

Finally, it is clear that there is potential in appropriate preprocessing and enrichment of text that is used any where in the pipeline. Good text enrichment has always improved precision, recall and FEVER scoring of any ER-CV pair. In this study however, no ER-CV pairs outperformed the benchmark on the FEVER Scores, we acquire solutions that could rival the state of the art, since we outperformed the benchmark in the ER sub tasks leading up to the main FEVER scoring task.

For future work, we could consider some Network-based or Knowledge Graph based approaches including Ciampaglia *et al.* (2017) approach to Fact-checking by transforming a sentence into a subject-object-predicate triple and checking the probability of an observed triple being in a graph through path semantic proximity between entities under a transitive closure [28]. For this approach a distantly supervised relation extraction method for text will be required to describe the relationship between identified entities into a Knowledge Graph [7].

Appendix A

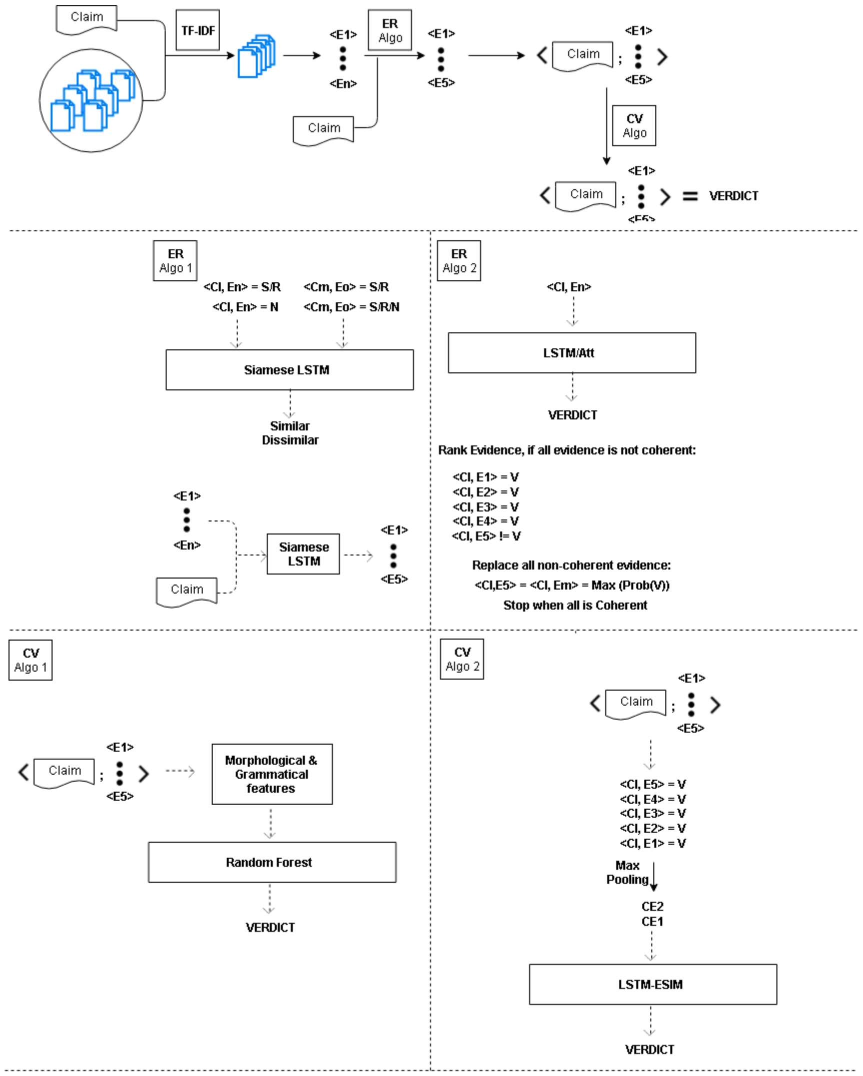

FIGURE A.1: This figure depicts our fact-checking process and methodology. The first lane is pipeline flow as an overview. ER Algo(1,2,3) and CV Algo(1,2) implementations are substituted as ER Algo and CV Algo respectively.

Bibliography

[1] X. Zhou, R. Zafarani, K. Shu, and H. Liu, "Fake news: Fundamental theories, detection strategies and challenges", WSDM '19, 836–837, 2019. DOI: 10.1145/3289600.3291382. [Online]. Available: https://doi.org/10.1145/3289600.3291382.

[2] E. Matsa, *The impact of fake news: Society*, https://www.kingsleynapley.co.uk/insights/blogs/criminal-law-blog/the-impact-of-fake-news-society, Accessed: 2019-11-30, 2019.

[3] A. Roy, K. Basak, A. Ekbal, and P. Bhattacharyya, "A deep ensemble framework for fake news detection and classification", *CoRR*, vol. abs/1811.04670, 2018.

[4] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection", English, in *Proceedings of The 12th Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, May 2020, pp. 6086–6093, ISBN: 979-10-95546-34-4. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.747.

[5] Á. Figueira, N. Guimarães, and L. Torgo, "Current state of the art to detect fake news in social media: Global trendings and next challenges.", in *WEBIST*, 2018, pp. 332–339.

[6] V. Kevin, B. Högden, C. Schwenger, A. Şahan, N. Madan, P. Aggarwal, A. Bangaru, F. Muradov, and A. Aker, "Information nutrition labels: A plugin for online news evaluation", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 28–33. DOI: 10.18653/v1/W18-5505. [Online]. Available: https://www.aclweb.org/anthology/W18-5505.

[7] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions", in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3346–3359. [Online]. Available: https://www.aclweb.org/anthology/C18-1283.

[8] 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)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 809–819. DOI: 10.18653/v1/N18-1074. [Online]. Available: https://www.aclweb.org/anthology/N18-1074.

[9] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection", pp. 422–426, Jul. 2017. DOI: 10.18653/v1/P17-2067. [Online]. Available: https://www.aclweb.org/anthology/P17-2067.

[10] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, "Finding streams in knowledge graphs to support fact checking", in *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2017, pp. 859–864.

[11] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning", *CoRR*, 2015.

[12] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with Siamese recurrent networks", in *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 148–157. DOI: 10.18653/v1/W16-1617. [Online]. Available: https://www.aclweb.org/anthology/W16-1617.

[13] N. Naderi and G. Hirst, "Automated fact-checking of claims in argumentative parliamentary debates", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 60–65. [Online]. Available: https://www.aclweb.org/anthology/W18-5509.

[14] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries", in *Proceedings of the first instructional conference on machine learning*, Piscataway, NJ, vol. 242, 2003, pp. 133–142.

[15] E. D. Liddy, "Automatic document retrieval in encyclopedia of language and linguistics", vol. 2, Elsevier Press, May 2005.

[16] J. Pennington, S. Richard, and M. Christopher D, *Glove: Global vectors for word representation*, 2014. [Online]. Available: \url{https://nlp.stanford.edu/projects/glove/}.

[17] Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method", *CoRR*, 2014.

[18] O. Papadopoulou, M. Zampoglou, S. Papadopoulos, and I. Kompatsiaris, "A two-level classification approach for detecting clickbait posts using text-based features", *CoRR*, 2017.

[19] D. Esteves, A. J. Reddy, P. Chawla, and J. Lehmann, "Belittling the source: Trustworthiness indicators to obfuscate fake news on the web", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 50–59. DOI: 10.18653/v1/W18-5508. [Online]. Available: https://www.aclweb.org/anthology/W18-5508.

[20] D. Caled and M. J. Silva, "Ftr-18: Collecting rumours on football transfer news", 2018.

[21] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, and I. Gurevych, "UKP-athene: Multi-sentence textual entailment for claim verification", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 103–108. DOI: 10.18653/v1/W18-5516. [Online]. Available: https://www.aclweb.org/anthology/W18-5516.

[22] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training", *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[23] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The fact extraction and VERification (FEVER) shared task", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 1–9. DOI: 10.18653/v1/W18-5501. [Online]. Available: https://www.aclweb.org/anthology/W18-5501.

[24] M. Taniguchi, Y. Miura, and T. Ohkuma, "Joint modeling for query expansion and information extraction with reinforcement learning", in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 34–39. DOI: 10.18653/v1/W18-5506. [Online]. Available: https://www.aclweb.org/anthology/W18-5506.

[25] H. Bahuleyan and O. Vechtomova, "Uwaterloo at semeval-2017 task 8: Detecting stance towards rumours with topic independent features", in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 461–464.

[26] C. Malon, "Team papelo: Transformer networks at FEVER", pp. 109–113, Nov. 2018. DOI: 10.18653/v1/W18-5517. [Online]. Available: https://www.aclweb.org/anthology/W18-5517.

[27] Y. Nie, H. Chen, and M. Bansal, "Combining fact extraction and verification with neural semantic matching networks", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6859–6866.

[28] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks", *PloS one*, vol. 10, no. 6, 2015.