ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

NCSR DEMOKRITOS
INSTITUTE OF INFORMATICS AND
TELECOMMUNICATIONS

Δι-ιδρυματικό Πρόγραμμα Μεταπτυχιακών Σπουδών στην «Τεχνητή Νοημοσύνη»

ΒΕΒΑΙΩΣΗ ΕΚΠΟΝΗΣΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

«Δηλώνω υπεύθυνα ότι η συγκεκριμένη Διπλωματική Εργασία για τη λήψη του μεταπτυχιακού τίτλου σπουδών του ΔΠΜΣ «Τεχνητή Νοημοσύνη» έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει εγκριθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

Η εργασία αυτή έχοντας εκπονηθεί από εμένα, αντιπροσωπεύει τις προσωπικές μου απόψεις επί του θέματος. Οι πηγές στις οποίες ανέτρεξα για την εκπόνηση της συγκεκριμένης Διπλωματικής αναφέρονται στο σύνολό τους, δίνοντας πλήρεις αναφορές στους συγγραφείς, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται στον Κανονισμό Σπουδών του Διιδρυματικού Προγράμματος Μεταπτυχιακών Σπουδών και στις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας».

Ο ΔΗΛΩΝ

Ονοματεπώνυμο: Γκάτσης Βασίλειος

Αριθμός Μητρώου: MTN 2004

Υπογραφή:

ΒΕΒΑΙΩΣΗ ΕΛΕΓΧΟΥ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η συγκεκριμένη Διπλωματική Εργασία για τη λήψη του μεταπτυχιακού τίτλου σπουδών του ΔΠΜΣ «Τεχνητή Νοημοσύνη» έχει ελεγχθεί από το σύστημα TurnItIn που παρέχεται από την Βιβλιοθήκη του Παν. Πειραιώς και η σχετική αναφορά του συστήματος (αποτελέσματα ελέγχου) είναι ως εξής:

18% ομοιότητα.

Ο ΔΗΛΩΝ

Ονοματεπώνυμο: Γκάτσης Βασίλειος

Αριθμός Μητρώου: ΜΤΝ 2004

Υπογραφή:

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

NCSR DEMOKRITOS
INSTITUTE OF INFORMATICS AND
TELECOMMUNICATIONS

A comparative study on explainable machine learning models for fact checking.

by

Vasileios Gkatsis

Submitted

in partial fulfilment of the requirements for the degree of

Master of Artificial Intelligence

at the

UNIVERSITY OF PIRAEUS

February 2022

*University of Piraeus, NCSR “Demokritos”.

Author . . . . . Gkatsis Vasileios. . . . . .

II-MSc “Artificial Intelligence”
February, 2022

Certified by. . . . . .

Georgios Giannakopoulos
Researcher at NCSR ’Demokritos’
Thesis Supervisor

Certified by. . . . . .

Vassiliki Rentoumi
Researcher
Member of Examination Committee

Certified by. . . . . .

Georgios Petasis
Researcher
Member of Examination Committee

A comparative study on explainable machine learning models for fact checking.

By

Vasileios Gkatsis

Submitted to the II-MSc "Artificial Intelligence" on

February 2022,

in partial fulfillment of the

requirements for the MSc degree

Abstract

Fact checking, as the task of assessing the validity of a claim or a piece of news, is a very important process for both journalists and public, especially in the era of social media. A huge amount of research has been addressed towards finding automated solutions for this problem. The recent advancements on Artificial Intelligence and Machine Learning have provided tools and frameworks with very good results. Especially with the recent enhancements of hardware, the development of state-of-the-art algorithms and mostly the availability of high quality data, tremendous progress has been made. With the broader use of such methods the request for reliability has began to emerge. That means that models should not appear as black boxes but their actions should be clear and understandable by humans. The two terms which describe that need are *interpretability* and *explainability*. Interpretability can be viewed as the ability of a machine learning model's actions to be transparent, while explainability is the ability of the model to use human understandable means of providing explanations about it's actions. Different approaches have been proposed in order to achieve such models, and discussions have risen on the usefulness of certain methods. In this thesis we study two

different explanation approaches. One uses a set of words that helped the most in in the fact checking process, and the other uses short summaries extracted from ruling articles. Then we propose a new high level taxonomy of claim justifications which can serve as an evaluation method for the aforementioned approaches as well as for a new means of explanation.

Thesis Supervisor: George Giannakopoulos
Title: Researcher at NCSR 'Demokritos'

Περίληψη

Η διασταύρωση ισχυρισμών, η οποία ορίζεται ως η διαδικασία αξιολόγησης της εγκυρότητας ενός ισχυρισμού ή μίας είδησης, είναι μια πολύ σημαντική διαδικασία τόσο για τους δημοσιογράφους όσο και για το κοινό, ιδιαιτέρως στην εποχή των μέσων κοινωνικής δικτύωσης. Μεγάλο κομμάτι έρευνας έχει ως στόχο του την δημιουργία αυτοματοποιημένων λύσεων και εργαλείων που αφορούν στη διασταύρωση ισχυρισμών. Η πρόσφατη πρόοδος στον τομέα της Τεχνητής Νοημοσύνης και ιδιαίτερα της Μηχανικής Μάθησης βοήθησαν στη δημιουργία τέτοιον εργαλείων και μάλιστα με πολύ καλά αποτελέσματα. Οι βελτιώσεις στον τομέα του υλικού των ηλεκτρονικών υπολογιστών, η ανάπτυξη μοντέρνων αλγορίθμων αλλά κυρίως η συγκέντρωση και δημόσια διάθεση υψηλής ποιότητας δεδομένων, βοήθησαν ιδιαίτερα σε αυτή την πρόοδο. Βέβαια, με την διάδωση της χρήση τέτοιων μεθόδων ανέκυψε και το ζητούμενο της αξιοπιστίας. Τα μοντέλα που αναπτύσσονται γι' αυτό τον σκοπό θα πρέπει να εγκαταλείψουν την κλασική δομή του 'μαύρου κουτιού' και να υιοθετήσουν πιο διαφανείς μεθόδους έτσι ώστε οι δράσεις και οι επιλογές τους να μπορούν να γίνουν εύκολα αντιληπτές από τον άνθρωπο. Οι δύο έννοιες που περιγράφουν αυτή η ανάγκη είναι, η *ερμηνευσιμότητα* και η *εξηγησιμότητα*. Η ερμηνευσιμότητα περιγράφει την ικανότητα ενός συστήματος να παρέχει διαφάνεια ως προς τις δράσεις του, ενώ η εξηγησιμότητα περιλαμβάνει το μέσο που χρησιμοποιείται για να γίνουν αντιληπτοί ο λόγοι οι οποίοι οδήγησαν σε αυτές τις δράσεις. Οι ερευνητές έχουν προτείνει διαφορετικούς τρόπους για να δημιουργήσουν μοντέλα που περιλαμβάνουν αυτές τις ιδιότητες, ενώ ταυτόχρονα γίνονται συζητήσεις ως προς την χρησιμότητα κάποιων εξ αυτών. Στην παρούσα εργασία μελετάμε δυο διαφορετικές μεθόδους. Η μία χρησιμοποιεί ως μέσο εξήγησης ένα σύνολο από λέξεις που βοήθησαν περισσότερο το μοντέλο να καταλήξει στο συμπέρασμα της διασταύρωσης του ισχυρισμού, και η άλλη χρησιμοποιεί μικρές περιλήψεις από άρθρα διασταύρωσης ισχυρισμών. Στο πλαίσιο αυτής της μελέτης

προτείνουμε τη δημιουργία μιας νέας ταξονομίας τόσο για την αξιολόγηση των παραπάνω μεθόδων αλλά και ως ένα νέο μέσο εξηγησιμότητας.

Acknowledgments

First and foremost I would like to thank my supervisor Dr Georgios Giannakopoulos and Dr Nikiforos Pittaras, for their continuous support, guidance and advising through all this period that I have been working on my masters thesis. Their help was absolutely crucial in order for it to be completed. I want also to thank Dr Aris Kosmopoulos for his guidance and for providing me with a base for my annotation tool.

I would also like to thank Dr Kashyap Popat and Dr Pepa Atanasova for providing me with the code of their scientific researches. Their quick and meaningful responses to my questions where a huge boost for the progression of this work.

I should also thank Sofoklis Karavellas for giving me access to the hardware resources of NCSR Demokritos.

Finally I want to thank all those who have supported me for the last one and a half year that I have been studying on the Artificial Intelligence masters program. These are my parents, Elpida and Dionysis, my sister Zina, my closest friends , Evi, Giorgos, Michaela and Stellina, and of course my girlfriend Georgia who has given me more support than what I could have imagined.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Fact checking, defined as the process of assigning a truth value to a claim made within a particular context[1], is a very crucial task in the field of news. Journalists have been fact checking manually for a long time, in order to determine whether a piece of news is trustworthy or not. Table 1 shows two claims which have passed the fact checking process and have been assigned a veracity label.The introduction and widespread of the internet though, as well as the massive growth of social media platforms, have changed the way information flows and have rendered the manual methods less useful. News can now travel from one side of the world to another in a matter of seconds, and along with real ones, fake news have also observed an increase in their propagation speed. Thus automated methods for fact checking have become an essential need for both journalists and readers.

Table 1: Examples of fact checked claims from LIAR dataset.

Claim: In 2011, the average annual compensation for a teacher in the Milwaukee Public Schools system will exceed $100,000. **Veracity Label:** True
Claim: The Chamber of Commerce says new carbon regulations will kill 244,000 jobs a year and cost average families $1,200 a year. **Veracity Label:** False

The evolution of Artificial Intelligence and machine learning have provided the necessary tools in order to create such methods, capable of replacing the ones that where used previously. Researchers working on machine learning, fact checking models have provided very useful results. Especially since the beginning of the previous decade, it has become clear that it is possible to study the propagation of fake news on social media

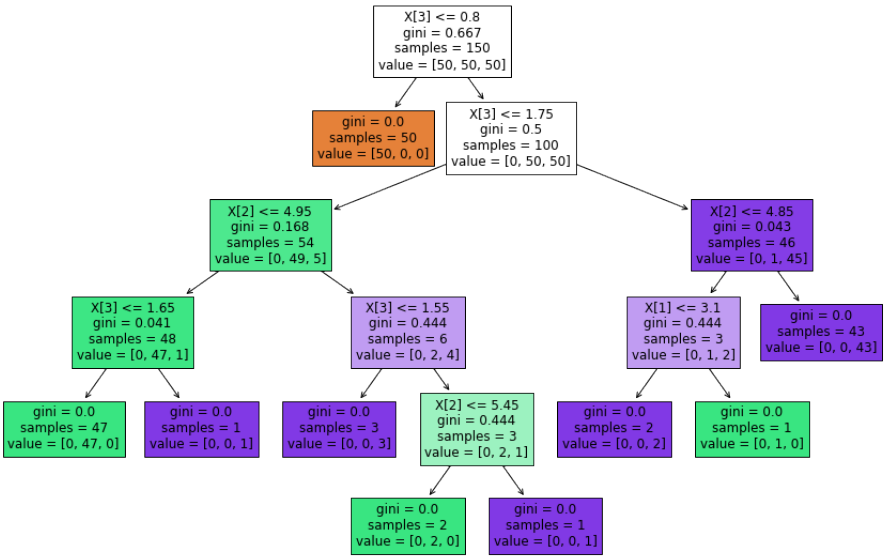
and even create tools with good capabilities on discriminating lies from truths [2][3]. Although progress has been continuous on this task, it is still not considered to be trivial. The recent development of both advanced hardware and state-of-the-art algorithms as well as the mass collection and publication of news data from reliable sources such as authorised fact checking websites[1][4], have given a push in this research field and have resulted in the creation of very accurate models.

Good predicting capability though, is not the only needed feature for such models. Fake news have proven to be very effective in manipulating human behaviour, collectively and individually, from tempering with election results to, in certain cases, 'triggering' criminal activity [5]. That aspect of fake news poses several ethical questions, and has shifted the research community's attention from *whether* a claim is true to *why* it is found to be true. The mere classification of a claim is not enough. It should be accompanied by a clear, reliable and easily received, by human intuition, justification, as to why the automated tool has assigned this specific label to a claim.

But what makes a fact checking tool reliable? The two key terms in order to answer this question are *Interpretability* and *Explainability*. These notions capture the researcher's interest more and more, but still there is not a consensus on how they should be received, and many tend to use them interchangeably. For the purposes of this thesis we are going to use the definitions provided by *Montavon et. al.*[6] which we consider to best describe the aforementioned terms. We consider a machine learning algorithm *Interpretable* if it allows for humans to clearly understand the processes that take place and how they lead to the final result. Decision Trees are an example of interpretable machine learning algorithm since the prediction process can be clearly depicted and recreated by a human. Figure 1 show an example of a decision tree.
We consider an algorithm *Explainable* if the means that are used for justifying the final results are clear and easily understood. Many such means have been developed and used by researchers, *Attention mechanisms*, *heat-maps* and *textual representations*

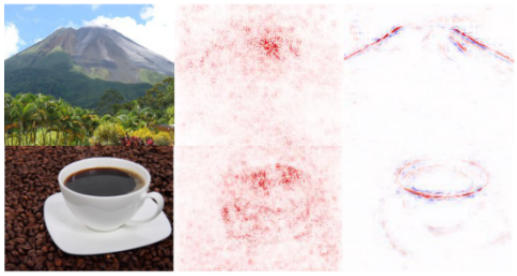
as shown in Figure 2, are just a few of them. Still there is no particular one considered to be best. Discussions have emerged as per the advantages and disadvantages of each one, with some methods getting more attention than others. Particularly attention based methods have recently been in the center of discussion as to whether they really provide meaningful explanations [7][8]. Guided by this discussion we decided to investigate ways for evaluating the efficiency of a justification method over another.

Figure 1: Plot of Decision Tree trained on the iris dataset.



[True] : Household paper shredders can pose a danger to children and pets.
Article Source: *byegoff.com*
packages while still protecting any private information that may be contained in the papers in theory the personal home paper shredder makes much sense personal or pet injuries from paper shredders a growing number of reported injuries reveal that home shredders pose a danger to any user and are especially dangerous to children and pets in fact the federal consumer product safety commission issued a paper shredder safety alert documenting reports of incidents involving finger amputations lacerations and other finger injuries directly connected to the use of home shredders

(a) Attention based justification.[9]



(b) Justification using SA and LRP heatmaps. [10]

Claim: The last major oil spill from a drilling accident in America happened over 40 years ago in 1969.

Ruling Comments: (...) The last major oil spill from a drilling accident in America happened over 40 years ago in 1969.
(...) The largest in volume was the Santa Barbara spill of 1969 referenced by Murdock and Johnson, in which an estimated 100,000 barrels of oil spilled into the Pacific Ocean, according to the API. The Santa Barbara spill was so big it ranked seventh among the 10 largest oil spills caused by marine well blowouts in the world, the report states. Two other U.S. spills, both in 1970, rank eighth and 10th. Fourteen marine blowouts have taken place in the U.S. between 1969 and 2007. Six of them took place after 1990 and spilled a total of nearly 13,700 barrels.
(...) We interviewed three scientists who said that the impact of a spill has little to do with its volume. Scientists have proven that spills far smaller than Santa Barbara's have been devastating.

(c) Textual Justification[11]

Figure 2: Three different justification methods.

1.2 Contributions

Our contribution on the research domain of explainable fact checking models is a new method for evaluating two different types of explanations. More specifically:

- We propose a new set of *abstract justification classes* which can be used as a means of explainability.
- We create a *new, justification* dataset containing claims annotated to our justification classes.
- We create two different pipelines that project a model's means of explanation to one of our justification classes. Thus we achieve evaluation by classification.
- We test our method using a model that returns a set of words which had the highest impact on the fact checking process, and another one that creates summaries from long explanatory articles.

- We compare the final results using statistical tests in order to find the statistical significance of our findings and we conclude that it is possible to evaluate the two methods via comparison.

1.3 Thesis Structure

The rest of this thesis is structured as follows. Section 2 (Background), contains an introduction of the basic terms and concepts that will be found in the paper, and gives an overview of the research domain of explainable fact checking algorithms, popular tools and methods as well as a short discussion on their efficiency. Then the problem that we are going to solve is defined mathematically. In Section 3 (Proposed Method) we describe our own method for solving the previously defined problem. The method is also formulated mathematically in order to match with the problem definition. We then provide an in depth analysis of each component of our method. In Section 4 (Experiments) we formulate our *null* hypothesis which we would like to reject. Then we describe the experimentation phase, the used dataset and the actions that have formulated it, the experimental setup that we have used and finally the results that lead to the rejection of our null hypothesis. The final section is Section 5 (Conclusions) where we present our findings. We initialise a short discussion on them, and then we propose some key steps that we believe could improve our work in the future.

2 Background

2.1 Main Terms and Concepts

Before we proceed into more specific parts of this thesis, it is important to describe the most important terms and concepts that will be used. Our intention, for this section, is to equip the reader with all the needed knowledge so that the reading of the rest of this work will be continuous and uninterrupted.

2.1.1 Machine Learning

Machine learning (ML) has been a research domain for many decades now, and it has always been the field that covers the attempts of humans to create intelligent computer systems. It has thus been the essence of Artificial Intelligence. There are two popular definitions concerning ML. One given by *Arthur L. Samuel* in 1959 which says that

”Machine learning is the field of study that gives computer the ability to learn without being explicitly programmed”[12].

The second one comes from 1997 and *Tom M Mitchell* who said that

”Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience.”[13]

The second definition particularly is very close to what ML represents for today’s scientists. Although there has been a huge increase in research in the last decades, partially due to the hardware and software improvements but mainly because of the huge available amounts of data, still ML is the field of computer science that aims to make computers auto-evolve on automated tasks. *Speech recognition*, *face recognition*, *recommendation systems*, are some of the tasks that ML is used for. The task of fact checking, on which we focus here, has seen dramatic improvements due to the use of ML,

especially in the last decades with the widespread of the internet and the social media platforms.

Machine Learning methods can be divided into for big subcategories: *Supervised learning*, *Unsupervised learning*, *Semi-Supervised learning* and *Reinforcement learning*[14].
In supervised learning the computer is trained on both data and the expected output. The aim of the training is to find a connection between the data and the output such that when new data are going to be inserted the model will produce the correct output. There are two big categories of supervised learning, *classification* and *regressions*, which will be analyzed later in this section.
Unsupervised learning is the exact opposite. The models in that category are trained with plain data and are expected to find some useful representation by themselves.
Semi-supervised learning is a combination of the two aforementioned processes where unlabeled data can be used to augment labeled datasets.
Finally reinforcement learning includes the methods that train a model by indications instead of data. The model has no knowledge of the world that surrounds it except from a predefined set of actions. For each action an indication as to whether it is good or bad is provided. The next thing for the model is to discriminate between sequences of 'good' actions against those of 'bad' actions.

2.1.2 Regression

As already said, regressions is a type of supervised learning. The characteristic of this method is that the output is given on *continuous* values. Hence it is recommended for problems where the need is to predict the specific value of a quantity based on a set of features. Some popular problems solved by regression methods are *stock value prediction* based on features such as current stock value, inflation etc., prediction of a *house's price* where the features could be the house's size, and location and *employ*

salary prediction with age and position as features. Some very well known regression machine learning algorithms are *linear*, *ridge* and *lasso* regression.

2.1.3 Classification

In Classification methods the predictions are given in *discreet* values also called classes. A very common example of machine learning classification algorithm is *spam detection* which is used by all the major email providers. The model is trained on spam and non-spam emails and tries to find the relation between the features and the class. Possible features can be the email length, the text-to-hyperlinks ratio etc. When the model is trained and applied, it automatically checks incoming emails, classifies them in one of two categories and if it considers the email to be spam it sends it to the spam folder.
In this work we are going to use classifiers as part of our proposed method. We are going to research the possibilities of classifying pairs of false claims and their justifications, into abstract justification classes.

2.1.4 Natural Language Processing

Natural language processing (NLP) is the domain of computer science that focuses on the process and analysis of human languages independently of the form (written or oral). Along with ML, it has been a topic of interest since the 1950s, although it did not always represent the same thing. Early research focused more on methods that would 'understand' a language but as time passed and the complexity of this task was revealed, researchers started focusing more on applicable tasks that would find hidden relationships between the different lexical phenomena. A good definition of NLP, given by *Elizabeth D. Liddy*, is this:

> "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more

levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. ”[15]

Nowadays ML is widely used for NLP task. In the last decades, the growth of ML have given a huge boost on performance of NLP applications, in task like *automatic summarisation*, *machine translation* ,*natural language understanding* and *fact checking*.

The work we present on this thesis is a part of the NLP domain. That is because firstly we study fact checking models, that use NLP techniques in order to classify the claims and to provide their explanations, and secondly the method we propose consists of creating algorithms that attempt to classify natural language text to more abstract classes.

2.1.5 Deep Learning

Deep Learning (DL) is a subset of machine learning. The difference between the two is that in contrast to ML methods, where we apply one process that consists of a simple or more complex algorithm, in DL methods, multiple processes work together, forming layers, in order to capture the underlying representation of the data [16]. Deep Learning methods consist of Deep Neural Networks (DNN). A Neural Network (NN) consist of multiple levels of multiple processing nodes. Each level, called a layer, is responsible for learning different relations of the data. The first layer is called input layer, then there is a number of hidden layers and last is the output layer. Nodes between layers are connected to each other and information propagates from layer to layer when it meets certain criteria. The difference between simple NNs, and DNNs is the number of hidden layers. DNNs contain a big number of hidden layers and thus are able to better process information.

In our work we will be studying two deep learning frameworks and their explainability methods. These are described in detail in sections 3.2.1 and 3.2.2.

2.1.6 Attention Mechanism

Attention mechanism method is very popular in NLP tasks. It was first proposed by *Bahdanau et a.*[17], and it consists a method which helps a DNN 'understand' which tokens of an input stream have greater importance than others. A usual way for measuring the attention weight of each token is by calculating a weighted average of their mathematical representations. Attention mechanisms have been widely used in fields beyond NLP. Computer vision is such a scientific field where the addition of attention mechanisms to DNNs have provided significant increase on results.

The fact checking deep learning framework DeClarE which we will be studying, uses such an attention mechanism in order to determine which parts of an article are related to the claim being checked.

2.2 Related Work

In sections 1.1 and 2.1 we have introduced the terms of interpretability and explainability. These concepts tend to get more and more necessary in order to create viable and ethical A.I. That is why a lot of research effort has been addressed towards that domain and different approaches have been proposed. In their survey on explainable artificial intelligence *Došilović et. al.* [18], split these approaches in two main categories, *Integrated*(transparency based) and *Post-Hoc*. Integrated methods are a "privilege" of simpler models. Those are the ones of lower complexity that can be easily (or at least with a minimum amount of effort) understood by a human, and their course of actions can be conceived and repeated manually. Linear models and decision trees are two such examples. But this is not possible on more complex models like Neural Networks and Support Vector Machines. That is where Post-Hoc methods can provide explainability and interpretability. These methods do not require the model to be transparent, and make use of external tools in order to extract information. For example *Assche et. all.*

[19] managed to train a single decision tree that would imitate the work of an ensemble of decision trees thus clarifying the prediction process.
Visual and textual techniques are also a post-hoc means of explainability. *Zeiler et. all.* [20] in their research provided visualisation methods in order to interpret intermediate layers of a Convolutional Neural Network, while *Park et. al.*[21] have introduced a method for visualising deep neural network's attention mechanisms on image data. These approaches are also used on fact checking models.
Kotonya et. al.[22] in their survey on explainable fact checking models distinguishes three big categories of state-of-the-art models with emphasis to explainability. These are a) *Attention Based*, b) *Rule Discovery*, and c) *Summarisation*. All of these methods are post-hoc and hence can be applied on the deep learning models used for fact checking.

Attention based methods are those where the model provides explanations by visualising a neural networks attention weights. *Popat et. al*[9] and *Yang et. al.*[23] are two such examples where words with higher attention weights are given higher visual importance in order to indicate that they played a greater role in the models final decision.

Rule discovery is a completely different approach where horn rules and knowledge graphs are used to create explanations. *Gad et. al.*[24] and *Ahmadi et. al.*[25] provide two different frameworks which automatically create a set of rules for fact checking and then expose the facts that support or contradict the initial claim.

Finally summarisation is the method of extracting explanations in the form of text summaries which are easier for humans to understand. A state of the art work in this method has been provided by *Atanasova et. al.*[11] where long fact checking articles are transformed in summaries.

Recently, discussions have focused on whether attention can be reliable as a means of explanation, and several remarks have been made. *Serrano et. al.*[26] and *Jain et.*

al.[7] both agree that attention weights can be deceiving as they may be unrelated with other more concrete measures of importance such as feature importance. *Wiegreffe et. al.*[8] poses challenges to *Jain et. al.*'s work, and proposes tests on four different frameworks: a simple uniform-weights baseline, a variance calibration based on multiple random seed runs, a diagnostic framework using frozen weights from pre-trained models, and an end-to-end adversarial attention training protocol, resulting that it is possible to get meaningful explanations from attention methods. This discussion in particular has influenced our research on methods for comparing the effectiveness of fact checking explanations.

Our approach on this discussion has been different than the previously mentioned methods. Instead of focusing on one method of explanation, and then disproving its' efficiency, we focused on finding a way to compare the explainability of two independent methods. The process we designed performs evaluation by classification. More specifically we have created a set of abstract classes of justification, we conducted a human annotation process in order to classify our data into the new justification classes and finally we trained simple machine learning classifiers to perform the same task as the annotators. The final results of the classifiers are measurable (e.g. in terms of accuracy, f1 scores etc.) and hence we can come to a conclusion as to whether one method of explainability is good or at least if it is better than another.

The process that we designed does not only provide a new way to compare two different explainability methods but it gets proven, later in the thesis, that our justification classes can also be used as a new means of explainability. Thus we propose a new explanation classes for fact checking models, and in that context we also provide a corresponding dataset.

2.3 Problem Definition

In this research we attempt to compare a fact checking (FC), machine learning model that uses Attention Based explanations FC_{att} and another one using Summarisation methods FC_{sum}. The challenge of this task lies on the fact that the two different forms of explainability can not be directly compared. The attention based model takes as input a claim c from a set of claims C and returns a veracity label $v \in V$ for the claim, and a multitude of n words $w_1, w_2 ... w_n$ which is a subset of the set W of total words in the examined articles. These are the words with the highest attention weight scores.

$$FC_{att} : C \longrightarrow W$$

The summarisation model, is given a claim c from the same set of claims C, and returns the veracity label $v \in V$ of the claim, as well as a short summary s extracted from an extensive fact checking article S.

$$FC_{sum} : C \longrightarrow S$$

It is important to note here that in this task we are not interested in the veracity result of the models and that is why the set V does not appear in the mathematical notations of the models as functions.
It is clear that the resulting explanation sets W and S contain different elements and hence a direct comparison is impossible. The problem that we are going to solve can be described as follows:

Problem. *Find a set J and two functions f_1 and f_2 such that:*

$$f_1 : W \longrightarrow J$$

$$f_2 : S \longrightarrow J$$

So we have to create the new set of classes J such that it is meaningful to map the elements from S and W to it, and find the functions/classifiers that are able to do the mapping efficiently. This way we end up with a common set of results and hence comparison is possible. The method followed in order to solve the above problem is described in detail in the next section.

3 Proposed Method

3.1 Method Overview

In the previous section we have showed that there is no unique method for justifying the results of machine learning models, and hence, there is no trivial way to compare the different approaches. Our method proposes the creation of a new set of classes in order to achieve evaluation by classification. The new classes consist of abstract forms of explanation. Then for each explanation method, a function is learned that projects the corresponding results to the new set of classes. This way common ground is created and thus the results are now comparable. We will use the same mathematical notations as in section 2.3, and we consider the set J as the set of justification classes.

$$f_1 : W \longrightarrow J$$

$$f_2 : S \longrightarrow J$$

So f_1 is the function that is given the attention weights $w_1, w_2, ..., w_n \in W$ provided by FC_{att} and matches them with a justification class $j \in J$, and f_2 is the function that given the summary $s \in S$ provided by FC_{sum} matches it with a justification class $j \in J$. The results of these two functions f_1 and f_2 are now easily comparable.

Now that we have described our problem and method mathematically we can present the algorithmic pipelines.
For the fact checking model that provides explanation through attention weights we have chosen *DecLarE* by *Popat et. al.* [9], and for the one that returns summaries we have chosen the one provided by *Atanasova et. al.* [11] in the paper *Generating Fact Checking Explanations*. The claims come from the dataset *LIAR-PLUS* by *Alhindi et. al.*[27], and they have been mapped to the new set of justification classes through a

human annotation process. Finally we have created the machine learning classifiers that match the given explanations to the new justification classes. Visual representation of the pipelines are provided in Figure 3 and Figure 4.

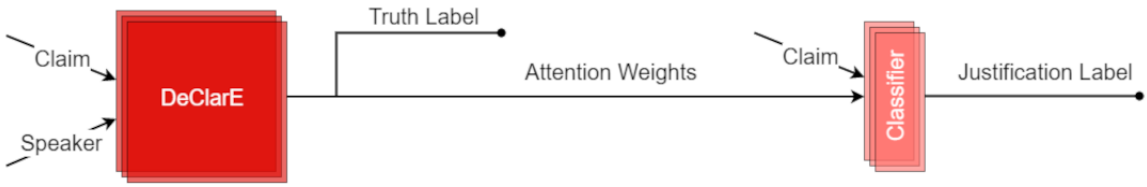


Figure 3: Attention Weights Pipeline

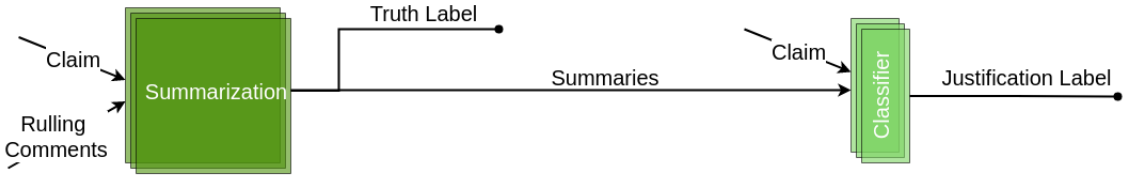


Figure 4: Summarisation Pipeline

3.2 Elaboration of Components

3.2.1 DeClarE

Popat et. al. in their work *"DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning"*[9], have created a deep learning fact checking framework which, provided a claim and it's originator, searches the web for relative articles, evaluates their credibility and returns a veracity label for the claim, in the set (True, False). It also returns a set of terms which assisted the most in the article credibility evaluation process and which are used as explanations for the models decision.

The framework uses a bidirectional LSTM[28] network in order to get the representation of each article (lower part of Figure 5). It also makes use of an attention mechanism, proposed by the authors, which ”computes the importance of each term in an article with respect to an overall representation of the corresponding claim.”[9] (upper part of Figure 5). This way they manage to capture the parts of the article that are most relevant with the given claim. An attention score is computed for each term of the article which can be viewed as the importance of the term. Then these terms that provided DeClarE with information on the articles credibility are the ones that are returned.

A human can easily understand and evaluate the relevance of these terms with the article and thus conclude whether the framework is accurate or not.

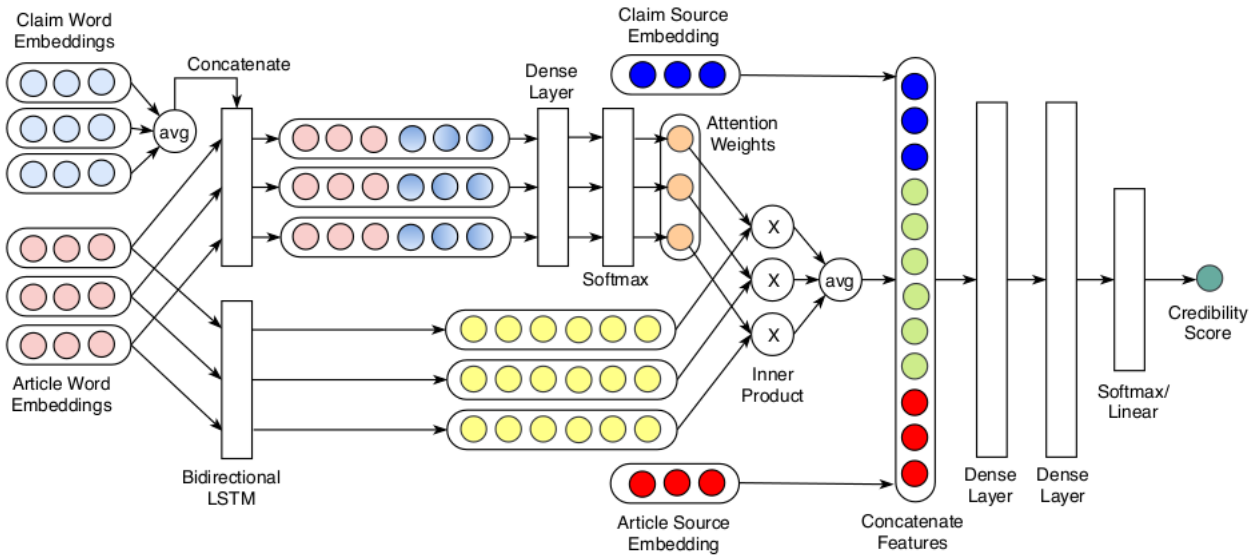


Figure 5: DecLarE sturcture as provided by *Popat et. al.*

3.2.2 Summarisation Model

Atanasova et. al. in their work *GeneratingFactCheckingExplanations"* [11] have created three models. One for explanation extraction, the second for fact checking and veracity prediction, and a third model which consists of the two previous models combined. For the creation of these models they have used the pre-trained *DistilBERT* model [29] which is a reduced version of BERT
BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained model that was designed to train deep bidirectional representations from unlabeled text. BERT is a *transfer learning* model as its usage consists of two phases. Phase one is pre-training where the model is given unlabeled data belonging to multiple tasks, and phase two is fine-tuning were the model is initialized with the pre-trained parameters and then each parameter is fine-tuned using labeled, task-specific data. BERT, and its transfer-learning approach have proven to be very usefull, especially in NLP tasks.
DistilBERT[29] model is a reduced version of BERT which was achieved by applying distillation techniques. Knowledge distillation, is a compression technique in which a compact model is trained to reproduce the behavior of a larger model[30][31]. By applying such techniques and without changing the original architecture of BERT model, *Sanh et. al.* have managed to achieve 40% reduction in terms of size, while retaining 97% of Bert's language understanding capabilities while being 60% faster. Figure 6 shows the structure of the explanation and fact checking model and Figure 7 shows the structure of the joint model. The explanation model creates an explanation for a given claim by attempting to maximize the similarity with a provided human justification. The fact checking model given the initial claim learns to optimise a cross-entropy loss function in order to predict the veracity label. The big contribution of this research is the proof that the joint model is capable of producing higher quality explanations than the sole explanation extraction model.

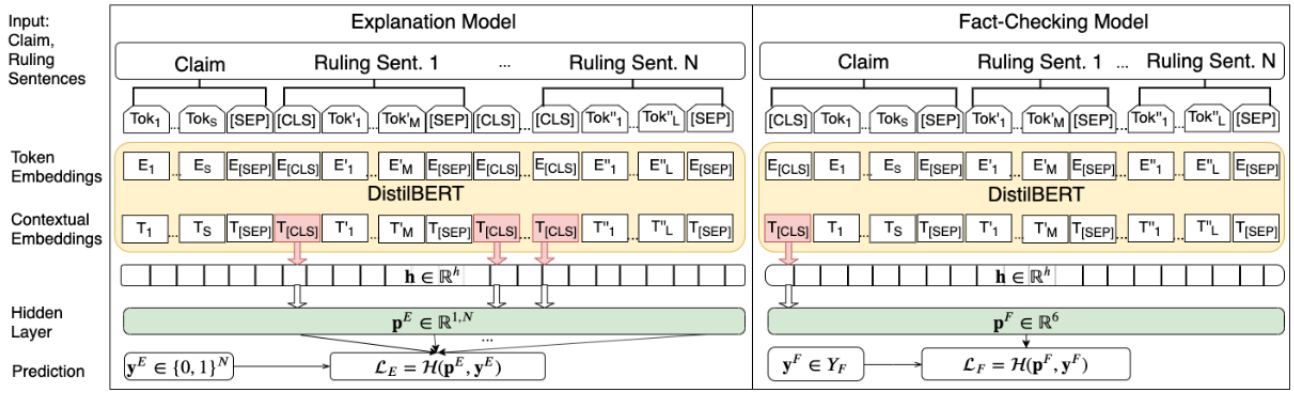


Figure 6: Structure of Explanation Model (Left) and Fact-Checking Model (Right)

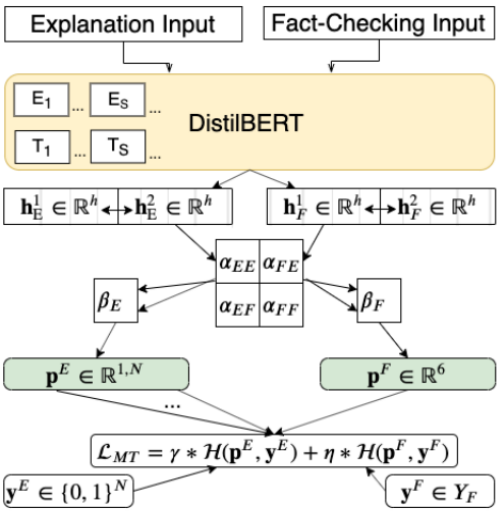


Figure 7: Joint Model

3.2.3 LIAR-PLUS

The LIAR-PLUS dataset by *Alhindi et. al.* [27] is an augmented version of the LIAR dataset by *Wang et. al*[4]. LIAR contains 12,836 short statements labeled for veracity(truthfulness), subject, context/venue, speaker, state, party, and prior history

statements taken from the political fact checking website *politifact.com* [1]. For the veracity of a claim one of six classes is chosen, which, ranked in ascending order of validity, are *Pants-on-Fire, False, Barely-True, Half-True, Mostly-True, True*. Each claim has been annotated by a professional editor, who provides an extensive explanatory article stating the claim, the facts and the reasoning that led him to assign the veracity label to this claim. Hence the overall validity of the dataset is greater than that of previous works which have focused on crowd-sourcing[32] [33].
The big contribution of the LIAR-PLUS dataset is that it has augmented the original LIAR dataset, by appending to each claim it's justification paragraph. This has been achieved by automatically extracting the last paragraph (usually named *"Our Ruling"*) of the article. In case this part did not exist at all then the last 5 sentences of the article where chosen.
Alhindi et. al.[27] proved that adding this piece of information along with the claim and other metadata could provide improved results in veracity classification task (both binary and six-class) independently of whether feature-based or deep learning models where used.

3.2.4 Justification Classes

In section 2.3 and section 3.1 we have described set J, which is the set of classes that attention words and summaries will be mapped to. Members of the J set are the justification classes that we will describe in this section. We wanted these classes to represent the claim-justification pairs of the LIAR-PLUS dataset in a more abstract way. Specifically we wanted the classes to capture the essence of why a claim is labeled as *false*. In order to achieve this we studied the dataset extensively and we have also studied the theory of *Rhetorical Devices*[34].

[1] www.politifact.com

3.2.4.1 Studying the Dataset

In all machine learning tasks regardless, one of the very first steps towards the solution is studying and understanding the dataset. This process is crucial as it provides the researcher, with important information concerning the structure of the dataset, obvious similarities and differences between entries, and even peculiarities of the data. This information create a very first image of the dataset.
In our case the goal was to create the new set of abstract justification classes, that will be used for evaluation by classification. So studying the dataset meant spotting more or less important semantic differences between different claims and their respective justifications, and thus forming a first version of the J set classes.
Although understanding the data is a very important step, it is not sufficient by itself in order for the new classes to be valid and scientifically stable. For that reason we have also studied the theory of *Rhetorical Devices* as described in the next section.

3.2.4.2 Rhetorical Devices

Rhetorical devices is a set of known mechanisms used by public speakers or writers in order to persuade the public about their claims. One of the most difficult tasks for a public speaker is to bridge the gap between their own view of the world and that of the audience. Rhetorical devices are a way to manipulate the thoughts and emotions of the receiving end, and make them embrace their ideas. [35]. *Robert A. Harris* on his work *A Handbook of Rhetorical Devices*[34] provides an extensive list of such devices, mainly focusing on three characteristics:

- Devices involving emphasis, association, clarification, and focus.
- Devices involving physical organization, transition, and disposition or arrangement.

- Devices involving decoration and variety.

In his work each Rhetorical device is thoroughly explained using both definitions and real examples of its usage. Examples of some of the devices that helped us most are presented in Table 2.
Since in our work we had to deal with false claims, rhetorical devices proved to be a valuable tool in terms of digging deeper into the ways that people manipulate the language for their purposes, and hence it was easier to discover distinct differences between false claims that would form our final justification classes.

Table 2: Sample dataset rows.

Device	Definition	Example
Understatement	Deliberately expresses an idea as less important than it actually is, either for ironic emphasis or for politeness and tact.	The 1906 San Francisco earthquake interrupted business somewhat in the downtown area.
Hyperbole	Deliberately exaggerates conditions for emphasis or effect.	There are a thousand reasons why more research is needed on solar energy.
Catachresis	An extravagant, implied metaphor using words in an alien or unusual way.	I will speak daggers to her. *(Hamlet)*
Apophasis	Asserts or emphasizes something by pointedly seeming to pass over, ignore, or deny it.	If you were not my father, I would say you were perverse. *(Antigone)*

3.2.4.3 Creating the classes

After finished studying the above mentioned methods we were able to create our set of justification classes. The first version of the justification set contained *five* different classes, which were:

Distortion: The speaker changes a known, or commonsense truth in order to support their claim.

Emphasis: The speaker selects and emphasizes on a single fact in order to prove their statement, although a lot more exist that disprove it.

Exaggeration: The speakers selects a single fact from a complex truth and augments it.

Unfounded: The speakers claim may or may not seem real but there is no fact to support it.

Vagueness: The speaker uses general phrases in no particular context in order to disorient the listener from the truth.

In order to test the clarity and descriptive effectiveness of the new classes we run a pilot annotation process using a small number of claim-justification pairs and three human annotators. The annotators were asked to describe how easy it was for them to assign each pair into one of the five classes. The annotators gave us their feedback which mainly focused on the fuzziness of selecting between the *Emphasis* and *Exaggeration* classes as well as the *Vagueness* class being rarely used. Taking their comments into consideration we have refactored the justification classes from *five* to *three* by merging the *Emphasis* and *Exaggeration* classes and removing the *Vagueness* class. The final form of the justification classes which was used for the annotation and the evaluation process is described below. Example pairs of claims and justifications for each class can be found in Table 3.

Distortion: The speaker changes a known, or commonsense truth in order to support their claim.

Emphasis: The speaker augments the importance of a single fact, which supports his position, although the truth is more complex.

Unfounded: The speakers claim may or may not seem real but there is no fact to support it.

Table 3: Examples from the dataset for Justification Classes Distortion, Emphasis, Unfounded (by order of appearance).

Claim: George Washington said a free people should be armed to guard against government tyranny. **Justification:** "George Washington said a free people should be an armed people," seemingly tracks Washingtons words to the nation: "A free people ought not only to be armed, but disciplined. "Contrary to Gohmerts characterization, though, Washington was not speaking about citizens arming themselves in case of government tyranny. Quite the opposite: The president and former general was calling for disciplined troops to fight on behalf of the government.
Claim: President Obama once said he wants everybody in America to go to college. **Justification:** We found 18 statements from Obama about people attending college. In the vast majority of the 18, Obama talked about making college a possibility or included the option of attending community colleges or vocational training instead.
Claim: President Barack Obama has said that everybody should hate the police. **Justification:** Giuliani said Obama has said "that everybody should hate the police. "Throughout all of his comments since August, when the latest unrest over racial disparities in the criminal justice system began, Obama has continuously encouraged working with police to find solutions and make change. He has also repeatedly emphasized the importance of law enforcement in communities of color and the fact that police officers have a dangerous job.

3.2.5 Justification Classifiers

The functions f_1 and f_2 that we have described in section 2.3 and section 3.1 are those that do the actual mapping of the means of explanation to the justification classes. In our pipeline these functions are represented by machine learning classifiers. We have experimented with several algorithms of different types and complexities. For example we have used both the simple *Logistic Regression* classifier, as well as the *Multinomial Naive Bayes* which is considered to be best for text classification tasks. We will discuss more about the selection of classifiers in the following sections. The list of used justification classifiers can be found in Table 4.

Table 4: Justification Classifiers

1	K- Nearest Neighbors
2	Decision Trees
3	Random Forest
4	Support Vector Machines
5	Linear Support Vector Machines
6	Gaussian Process
7	Ada-Boost
8	Naive Bayes
9	Multinomial Naive Bayes
10	Stochastic Gradient Descent

3.2.5.1 K-Nearest Neighbors

K-Nearest Neighbors is a very simple yet efficient classification algorithm. The main idea behind this algorithm is that after the model has been trained it contains a $d - dimensional$ numerical representation of the input data, where d is the number of features per input. Then for each new test input the algorithm finds the K most

similar inputs (neighbors). Similarity is measured by calculating the distance between the new input and the train data. A common distance metric is euclidean distance, which we also used in our experiments. After the K neighbors have been found the new input is assigned to the class that the majority of its neighbors belong to. Thus each new input is assigned to the class that contains the most similar data.

3.2.5.2 Decision Trees

A Decision Tree classifier is also a simple model that works on a very basic principle. During training, labeled input data enter the root node, and decision criteria are created. For each criterion a new decision node is split from the origin node, in the form of a tree. The last layer of decision nodes that do not further split, are called leaf nodes. These nodes represent the labels that we want to assign our data to. During the testing process a new input is inserted in the root node and then a path of the tree is traversed according to the result of the decision criteria on each branch. According to the last leaf branch the input is assigned to the corresponding label.
Decision tree classifiers are very easy to implement and interpret and that is why they consist a popular classification algorithm. They are though better used for binary classification, and usually they are not used for text based problems.

3.2.5.3 Random Forest

Random Forest is not an algorithm, but an ensemble of decision trees working independently on the same task. Each model produces a classification outcome for each input and the final result is taken by majority vote. The important fact about this structure is the independence between the Decision Trees. Since there is no connection in the way that they treat the problem, it is less likely for them to lean towards the same errors, and though some trees may end up making mistakes, some others will predict correct. Thus we achieve better classification results when using

random forest than when using single decision trees.

3.2.5.4 Support Vector Machines

Support Vector Machines(SVM) is an algorithm mainly used for classification but which can also be extended to regression problems. The main idea behind the algorithm is that, given $d - dimensional$ data it searches for a clear way to separate the different classes in d dimensions. If for example it is given 2-dimensional data it will search for the line that best splits the classes. The efficiency of the (SVM) algorithm comes from the next step.
In case the data are inseparable in the current $d - dimensional$ space, SVM applies the so called *Kernel Trick*. That is that it tries to map the data points to a higher level of dimensions $d+1, d+2, ..., d+n$ and then search if in that new space the data points are separable. In the previous 2-dimensional example, if the data where inseparable by a line then the algorithm would map them into a 3-dimensional space and search for a plane that divides them best etc. That technique is what makes SVM stand out from the classical machine learning classification algorithms. The algorithm that performs the mapping is called a *kernel* and there are multiple different implementations of it.

3.2.5.5 Linear Support Vector Machines

Linear Support Vector Machines (LSVM) is the same as the SVM algorithm that we described above, with Linear kernel function, which is described as:

$$f(X) = W^T \times X + b$$

where W is the weight vector that we want to minimize, X is the data that we are trying to classify, and b is the linear coefficient estimated from the training data.
It is clear that this function returns a linear splitting point, and that is why LSVM are used in problems where data are linearly separable. Text tasks, which contain

multiple features are usually such tasks.

3.2.5.6 Gaussian Process

Gaussian Process is a non-parametric classification method which similarly to the SVM is implemented by a kernel based algorithm. The selected kernel specifies the covariance function of the data, and hence controls how the data points are grouped. According to the way that the data are grouped the algorithm determines the label of the new input.

3.2.5.7 Ada-Boost

Ada-Boost (Adaptive Boosting) classifier much like Random Forest is an ensemble of classifiers. It combines multiple low-performing, traditional classifiers in order to achieve better final results. Ada-Boost trains each individual classifier in an iterative way using the same dataset. After each training it adjusts the weights of incorrectly classified instances in order to focus the training on the harder or more unusual data.

3.2.5.8 Naive Bayes

Naive Bayes is a probabilistic classification algorithm which makes use of the Bayes Theorem, and assumes that all features are independent. This assumption is why it is called 'naive', as in real world problems it is not usual of features to be independent. Yet, in the general case, the algorithm provides good results. Bayes Theorem gives the probability of y happening given that x has already occurred, and its mathematical represented as follows:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

In the classification algorithm y is the resulting class and x is any number of features available, so ti can also be represented as $x = \{x_1, x_2, ..., x_n\}$. So after a series of

transformation on the Base Theorem we can get the probability of a set of features belonging to a class. The class with the highest probability is the one selected.

3.2.5.9 Multinomial Naive Bayes

Multinomial Naive Bays is a variation of Naive Bayes classifier, in which features represent the frequency of appearance of a given term. Since this is a common representation in textual tasks, such as news classification, where the feature vector contains the number of appearances of each word in a document, Multinomial Naive Bayes is highly preferred in such tasks.

3.2.5.10 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is not an actual classifier. In fact it is implemented as a linear classifier such as logistic regression, or SVM, which is optimized by the SGD algorithm. So Stochastic Gradient Descent is one of the most popular methods for minimizing a loss function. As such it is used to better train the classifier when reading the training data.

3.2.6 Word Transformation

The justification classifiers mentioned above, take as input a textual form of justification, which can be a series of words in the Attention pipeline or a small paragraph in the summarisation pipeline. Since the classifiers can not recognize text, the justifications must be transformed into numerical data. In our experiments we have used two very common methods of such transformation. These are *Tf-Idf* and *Word2Vec*. In the following paragraphs we explain how these methods work.

3.2.6.1 Tf-idf

Tf-idf, which stands for *Term Frequency-Inverse Document Frequency* is one of the most commonly used metrics representing the importance of a term in a given document. It is used in a multitude of text related tasks, including text classification, which is the case that we are studying. Transforming the input text of a classifier using the Tf-idf method will inform the classifier of each terms importance, and so words with higher importance will be highly evaluated in the classification process. In order to explain how Tf-idf is calculated we will examine each part separately.
Term Frequency is the ration of appearances of a term in a document to the total number of words in the document. Bellow is the mathematical type.

$$Tf = \frac{Number\ of\ term\ appearances\ in\ document}{Number\ of\ All\ words\ in\ document}$$

Inverse Document Frequency on the other hand is a measure of how much information the term provides, meaning if it is common or rare across all documents. In order to compute it we get fraction of the total number of documents by the number of documents containing the term, in a logarithmic scale as shown in the next mathematical type:

$$Idf = \log \frac{Number\ of\ documents}{Number\ of\ documents\ containing\ the\ term}$$

The final Tf-idf value is computed as the product of the two above mentioned terms.

$$Tf - idf = Tf \times Idf$$

This way for each document we obtain a vector of the Tf-idf scores for each word found in all the documents. It is clear that words that do not appear at all in the specific document have a Tf-idf score equal to 0.

3.2.6.2 Word2Vec

Word2Vec [36] is a *shallow neural network* which has proven to be very effective on creating numerical representations of text, called word embeddings. It is defined as shallow because it only contains one hidden layer, in contrast to deep neural networks which contain a larger amount of them. In practice there are two versions of the model. One implementing the *Continuous Bag Of Words (CBOW)* algorithm and on implementing the *Continuous Skip-Gram Model*. The main difference between the two is that the first algorithms creates a model that is trained to predict a word from its context words, while the second model is trained to predict the probability of a word being present when an input word is present. So both models tend to capture the relation between the words of a corpus in a different way. After the models are trained we just keep the numerical weights of the input words which then become our numerical representation. Word2Vec has proven to be very efficient in creating word embeddings, especially as the number of words in the corpus gets larger.

3.2.7 Evaluation

As already mentioned our goal is to evaluate the effectiveness of each justification method, as a means of explanation, by performing evaluation by classification. This way we will also be able to assess the usefulness of our proposed justification method

as a means of explanation. In this section we provide some more technical details about the way we implemented our evaluation process.

As shown in figure 8 and figure 9 the justification method provided by each model is fed into the justification classifiers who assign it a justification label.

Ideally we would train and test the justification classifiers using different datasets in order to be able to acquire a better view of their results. Since we are using only one small dataset we will create ten ”different” datasets from it, and then we will apply a 10-Fold validation process.

First we have split the dataset into three parts (train, validation and test), and then we created 10 permutations of them in such a way that no entry of one split is contained to another. Then we performed 10 experiment cycles, with each cycle containing the train, validation and testing of the classifier. For each cycle c, $c \in \{1, 2, ..., 10\}$ we have gathered the *accuracy* and *macro F1* scores of the classifier.

After the experiments were completed we also gathered the *mean accuracy* and *mean macro F1*.

Then we discarded the models who’s results were closer to randomness. Since we have three justification classes, a mean accuracy score close to 33.33% means that the classifier has not acquired any useful information from the data and the returned results are similar to that of a random selection process.

As a final step we have selected the top-3 (in terms of mean accuracy and mean macro f1 scores) classifiers for each pipeline and their counterparts from the other pipeline and compared their results. In order to find if their is statistical significance in the performance results of the classifiers for the 10 experiments, and hence to prove that the results are not due to chance, we have conducted the wilcoxon statistical test.

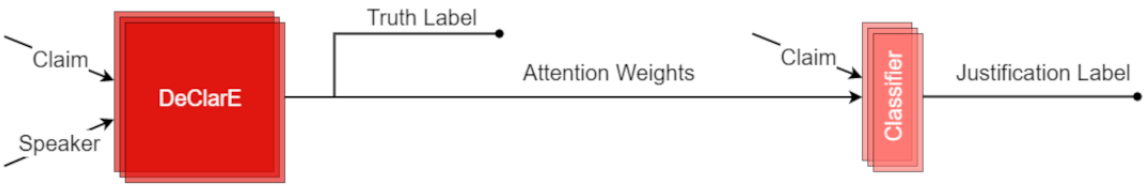


Figure 8: Attention Weights Pipeline

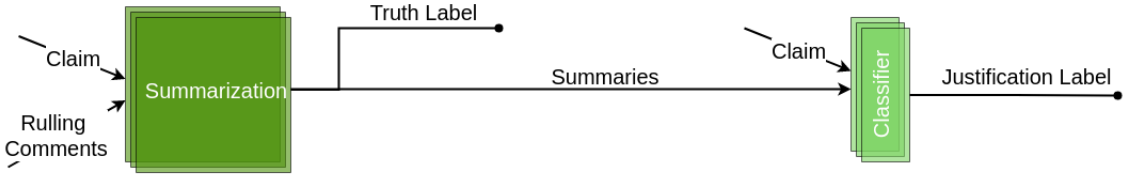


Figure 9: Summarisation Pipeline

4 Experiments

4.1 Hypothesis

In order to proceed with the experiments we first had to formulate our *null hypothesis* (H_0). Since we are dealing with a comparison of two subjects the possible ways are either to suppose that one of the two performs better, or that both perform equally good. In section 2.2 we have cited related work discussing the effectiveness of attention weights as a means of explainability. Following up on that work we will assume that attention methods compare worst than summaries in terms of explaining the veracity results of a fact checking model. Our *null hypothesis*, which we will try to *reject* is formulated as follows:

H_0: Explaining the results of a fact checking model using attention based methods is more accurate than using extracted summaries.

4.2 Data

For the experimentation phase we have created our own dataset. It contains a subset of the claim-justification pairs from LIAR-PLUS, while each pair has been assigned to one of the justification classes that we have described in section 3.2.4.

4.2.1 Veracity Label Merging

First of all we decided that it was meaningful to choose only *False* claims for out task. That decision was taken for two reasons: 1) Justifying a *True* claim does not have much space for ambiguity, meaning that a *true* claim is as close to the facts as possible. For that reason their is no need to map it to an abstract class of justification. 2) Our intended conclusion would be to evaluate the efficiency of each explainability

method, and explainability is mostly meaningful when it comes to *false* statements. Claims in the LIAR-PLUS dataset are assigned to one of six veracity labels ('pants-on-fire', 'false', 'barely true', 'half true', 'mostly true' and 'true'). In order to choose the false ones we needed to merge them in the two, basic, veracity classes *True* and *False*. We achieved this by using the following rule as a measure: if a claim is classified as 'true' or 'mostly-true' then we assign it to the *True* class. Else if the claim is classified as 'half-true', 'false', or 'pants-on-fire' we assign it to the *False* class. The discrimination is quite obvious except for the 'half true' label. Many arguments could occur as to which class it belongs to. Our approach is that a 'half true' claim leaves parts of the truth behind (maybe even crucial ones), and hence is closer to a lie. After the merging was completed. we have choose randomly *180* claim-justification pairs to instantiate our dataset.

4.2.2 Data-Model Compliance

The next step was to confirm whether our fact checking models could provide answers for the chosen claims. The summarisation model by *Atanasova et. al.*[11] is trained and tested on the LIAR-PLUS dataset, and so we already had the models response to our claims. On the other hand DeClarE by *Popat et. al.* which crawls the web for articles connected to the claim and extracts the most relevant snippets, could not manage to find answers for 23 out of the 180 claims. Hence our dataset was reduced to 157 pairs.

4.2.3 Annotation Process

The next step was the annotation process. We have created and online tool for annotating the claim-justification pairs into the justification classes. 13 annotators took part in this process and completed it. For each claim-justification pair the annotator would select the most suitable justification class or the 'Unclear' option in

case they were unable to decide. A screenshot from the annotation tool is provided in Figure 10.

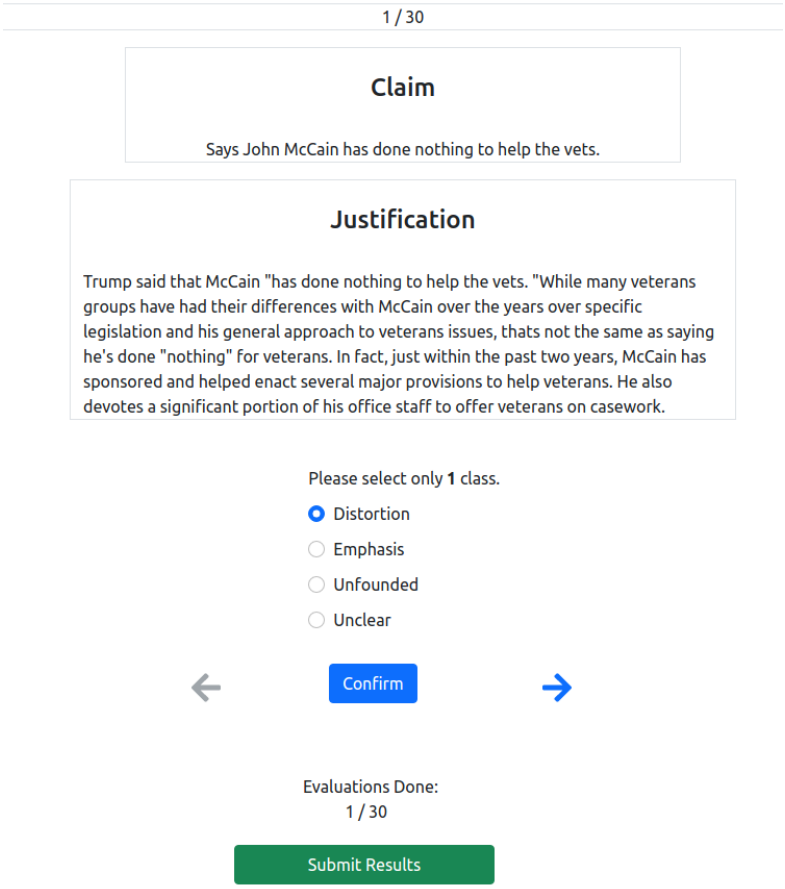


Figure 10: Screenshot from the annotation tool.

When all the annotators have finished, we gathered the results and applied a majority vote filter and a minimum 3 votes filter. This means that pairs with tied votes as well as pairs with less than 3 votes were discarded. The remaining pairs were assigned to the class with the majority of annotator votes. This process excluded 6 more claim-justification pairs and the final dataset was reduced to 151.
Last step was to check the balancing of classes in the dataset. The 151 pair dataset were slightly unbalanced as shown in Table 5a.

In order to balance it we applied an undersampling technique. We found the class with the lowest multitude of data, which was *emphasis*, and then we randomly selected and discarded such a number of data from the other two classes in order for them to have the same multitude as *emphasis* class. The final dataset contains *123* claims and is balanced as shown in Table 5b. A sample of the dataset is shown in Table 6.

Table 5: Balance of classes in justification dataset.

Class	# of data
distortion	63
unfounded	47
emphasis	41

(a) Unbalanced Dataset

Class	# of data
distortion	41
unfounded	41
emphasis	41

(b) Balanced Dataset

Table 6: Sample dataset rows.

Id	Claim	Justification Label
2828.json	Milwaukees problems include 52 percent black male unemployment.	unfounded
6162.json	Obama has racked up more debt than any president in history over $5 trillion.	distortion
10844.json	Every president since Franklin Delano Roosevelt has had fast-track trade authority.	emphasis

4.3 Experimental Setup

Our experimental setup consists of two justification pipelines, one for each fact checking model. These pipelines have the following structure:

- A fact checking model is given as input the claims from the justification dataset and returns a veracity label and a justification, in some model dependent form.
- The justification is then passed along with the claim as an input in the justification classifiers.

- The results from the justification classifiers of the two pipelines are compared and statisticaly tested in order to find which means of justification performs best.

Figure 4 depicts the first pipeline based on *DeClarE*, which we have described in section 3.2.1. We have used a pre-trained version of DeClarE provided to us by the author *Kashyap Popat* [9]. As already mentioned, in order to create the justification dataset we needed to test DeClarE on the selected claims. So by the time the dataset had reached the final form, we had already gathered the justification results of DeClarE for each claim, as a list of words.
For the second pipeline seen in Figure 5, the code for the model has been given to us by the author of the paper[11] *Pepa Atanasova*. We trained the summarisation model as described in *Section 4* of *"Generating Fact Checking Explanations"* [11] and then we fed the model with our 123 claims in order to get the resulting summaries.

4.3.1 Metrics

Before we move on to describe the experimental process for the justification classifiers we should first mention the metrics that we used for the experiments. These are *accuracy* and *macro F1*.

4.3.1.1 Accuracy

Accuracy is the simplest metric we can use and it is defined as:

$$Accuracy = \frac{Number\ of\ True\ predictions}{Number\ of\ All\ predictions}$$

Despite its simplicity accuracy gives a general overview of the performance of the classifier, and it can be used in order to optimize the classifier parameters.

4.3.1.2 Macro F1

F1 score is the harmonic mean between *precision* and *recall* and 'Macro' defines the averaging method. So in order to define macro F1 we must first define these two metrics.

Precision is a measure of the proportion of per class predictions that where actually correct.

$$Precision = \frac{Number\ of\ correct\ class\ predictions}{Number\ of\ All\ class\ predictions}$$

Recalls is the measure of proportion of per class predictions that were identified correctly.

$$Recall = \frac{Number\ of\ correct\ class\ predictions}{Number\ of\ All\ elements\ of class}$$

Finally F1 Score is a method that balances the results of precision and recall in order to provide a better understanding of the classifiers performance, and it is defined as:

$$F1\ Score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

The Macro averaging method is the simplest method for multi-class problems. And it is defined as the F1 score of each class divided by the number of classes. So in our case where we have three justification classes, *distortion*, *unfounded* and *emphasis* the macro f1 score is:

$$Macro\ F1\ score = \frac{F1_{distortion} + F1_{unfounded} + F1_{emphasis}}{3}$$

4.3.2 Train, Validate, Test

The last part of our experimental setup is the training, tuning and testing of the justification classifiers. We have split the dataset into three parts, one for training one for validating and one for testing, in such a way as for the validating and testing

datasets to be equal to 10% of the length of the original dataset. Since we intended to compare classifier results we needed the metric scores to be obtained from more than one experiment, or else we wouldn't be able to test if the final results are of statistical significant importance. For this reason we have created 10 permutations of the dataset structure that we mentioned above.
For each classifier the train, validate, test process was structured as follows:

- For each fold $i, i \in \{1, 10\}$ perform an inner K-Fold validation process for different sets of classifier parameters, and get the classifier mean accuracy in the inner K-Fold validation.
- Evaluate the classifier on the i-th validation dataset and get the accuracy score.
- Choose the classifier that maximizes both inner K-Fold and validation accuracy score.
- For each fold $i, i \in \{1, 10\}$ train the returned classifier on the i-th train dataset and test on the i-th test dataset.
- Collect the mean accuracy and mean macro F1 scores from the 10-Fold validation.

In order to transform our claims and justification into features we have experimented with *Tf-Idf* and *Word2Vec* methods. All the code for the dataset creation, data processing and experiments is uploaded and publicly available on github [2]. The results of this process are described in the next section.

[2] github.com/vGkatsis/MSc-Thesis

4.4 Results

In this section we present the results of our experimental setup as it has been described previously.

4.4.1 Inter-annotator Agreement

In order to evaluate the performance of the justification classifiers we would need a *lower* and an *upper* bound. Finding these two bounds will let as make comparisons between classifier performances. The metric that is commonly used as a lower bound, and which we will be using for our experiments is randomness, which we have already described in Section 3.2.7.
To get a notion of what an upper bound for classifiers accuracy and f1-scores could be, we have measured the *inter-annotator agreement*. In order to perform the measurements we gathered, for each annotator, the justification labels the they gave on the claim-justification pairs that they have been assigned, and calculated their accuracy and f1-scores based on the final labels that have been chosen through the majority-vote process. Finally we calculated the mean score for each metric. The results are shown on Table 7.

Table 7: Inter-Annotator Agreement

Metric	Mean Score
Accuracy	0.57
Macro F1	0.49

Before concluding with inter-annotator agreement it is important to make a key note concerning the nature of the annotation task. The annotators have been given pairs of claims and justification to assign the the justification labels. These claims are in the form of short paragraphs, similar to the ones that the summarisation model provides. So the annotators perform a task similar to the task of the summarisation pipeline

classifiers. In that case the inter-annotator agreement measure as described is a hard upper bound.
In the case of the attention pipeline the classifiers learn to assign justification labels to claims formed from key words and not short paragraphs. Hence the task is somewhat different, and the upper bound is not strict.
In a future work the annotators should be prompted to assign the same claims, with their justifications in the form of attention words, as provided by DeClarE, and two separate upper bounds should be created.

4.4.2 Top-3 Classifiers

After the 10-Fold validation has completed we have selected the top 3 classifiers from each pipeline by means of mean accuracy and mean macro f1 scores.
For the summarisation pipeline these classifiers are

- Gaussian Process Classifier
- K-Neighbors
- Multinomial Naive Bayes

For the attention pipeline the classifiers are:

- Decision Trees
- Gaussian Process Classifiers
- K-Neighbors

Their results in the 10-Fold validation are shown in Table 8 and Table 9. *Decision Trees* algorithm is one of the top-3 classifiers in the attention pipeline but not in the summarisation pipeline. For that reason we will still include it in the comparison regardless of it's results. The same applies for *Multinomial Naive Bayes* in

summarisation pipeline. We call these classifiers supplementary and their results are shown in Table 10.

4.4.3 Statistical Test

Finally in order to validate the results from the justification classifiers, we must find if they have significant statistical importance. Practically what we want to prove is that the dominance of one classifier over another in terms of mean macro f1 score and accuracy is not due to chance. For that reason we are going to use the *Wilcoxon signed-rank test*[37]. The wilcoxon test, tests the null hypothesis that two related paired samples come from the same distribution.
In our case we will give as input to the test the macro f1 and accuracy results of each classifier for each one of the 10 iterations, and we want to prove, with a confidence level of 5% that they do not come from the same distribution.
The results are shown in Table 11. We observe that for three out of four classifiers the wilcoxon p-values are greater than 5%. That means that the null hypothesis of the test is approved and we can not make any assumptions about them.
For the *Multinomial Naive Bayes* classifier though both p-values are 0.02 which means that the results of the classifier for the two pipelines do not come from the same distribution, and since the classifier achieved better results for the summarisation pipeline we can conclude that, at least for this sole classifier, the summarisation pipeline performs better than the attention pipeline, and thus reject our H_0 which stated that attention methods perform better as a means of explanations than extracted summaries.

Table 8: Top-3 Summarisation Pipeline Classifiers

(a) Gaussian Process Classifier (Tf-Idf)

Iteration	Accuracy	Macro F1
1	0.46	0.47
2	0.23	0.19
3	0.31	0.30
4	0.42	0.40
5	0.58	0.57
6	0.50	0.48
7	0.58	0.56
8	0.50	0.46
9	0.33	0.29
10	0.67	0.66
mean	0.46	0.44
standard deviation	0.130	0.137
standard error mean	0.044	0.046

(b) K-Neighbors (Tf-Idf)

Iteration	Accuracy	Macro F1
1	0.31	0.26
2	0.38	0.39
3	0.31	0.27
4	0.50	0.48
5	0.50	0.46
6	0.58	0.58
7	0.75	0.76
8	0.25	0.17
9	0.42	0.41
10	0.42	0.39
mean	0.44	0.42
standard deviation	0.140	0.161
standard error mean	0.047	0.054

(c) Multinomial Naive Bayes (Tf-Idf)

Iteration	Accuracy	Macro F1
1	0.38	0.40
2	0.31	0.31
3	0.23	0.23
4	0.42	0.40
5	0.58	0.57
6	0.50	0.49
7	0.58	0.56
8	0.58	0.54
9	0.17	0.13
10	0.58	0.54
mean	0.43	0.42
standard deviation	0.148	0.145
standard error mean	0.049	0.048

Table 9: Top-3 Attention Pipeline Classifiers

(a) Decision Trees (Word2Vec)

Iteration	Accuracy	Macro F1
1	0.46	0.44
2	0.54	0.53
3	0.38	0.38
4	0.58	0.59
5	0.50	0.50
6	0.50	0.48
7	0.08	0.07
8	0.42	0.41
9	0.33	0.31
10	0.58	0.56
mean	0.44	0.43
standard deviation	0.142	0.144
standard error mean	0.047	0.048

(b) Gaussian Process (Word2Vec)

Iteration	Accuracy	Macro F1
1	0.46	0.39
2	0.38	0.29
3	0.46	0.40
4	0.58	0.56
5	0.42	0.33
6	0.33	0.28
7	0.17	0.13
8	0.50	0.50
9	0.33	0.26
10	0.50	0.46
mean	0.41	0.36
standard deviation	0.110	0.121
standard error mean	0.037	0.040

(c) K-Neighbors (Word2Vec)

Iteration	Accuracy	Macro F1
1	0.46	0.46
2	0.46	0.39
3	0.15	0.18
4	0.58	0.59
5	0.33	0.34
6	0.42	0.34
7	0.25	0.21
8	0.33	0.28
9	0.50	0.48
10	0.33	0.18
mean	0.38	0.34
standard deviation	0.121	0.130
standard error mean	0.040	0.044

Table 10: Supplementary Classifiers

(a) Summarisation Decision Trees (Tf-Idf)

Iteration	Accuracy	Macro F1
1	0.23	0.21
2	0.23	0.20
3	0.31	0.31
4	0.58	0.58
5	0.50	0.49
6	0.25	0.25
7	0.58	0.58
8	0.17	0.14
9	0.25	0.22
10	0.25	0.17
mean	0.34	0.32
standard deviation	0.148	0.161
standard error mean	0.049	0.053

(b) Attention Multinomial Naive Bayes (Word2Vec)

Iteration	Accuracy	Macro F1
1	0.46	0.47
2	0.31	0.30
3	0.15	0.15
4	0.33	0.33
5	0.25	0.25
6	0.33	0.24
7	0.17	0.13
8	0.17	0.18
9	0.17	0.16
10	0.25	0.23
mean	0.26	0.25
standard deviation	0.094	0.097
standard error mean	0.031	0.033

Table 11: Wilcoxon Statistical Test Results (Attention vs Summarisation Model)

(a) Results for Macro F1

Classifier	Wilcoxon P-value
Gaussian Process Classifier	0.32
K-Neighbors	0.31
Multinomial Naive Bayes	0.02
Decision Trees	0.08

(b) Results for Accuracy

Classifier	Wilcoxon P-value
Gaussian Process Classifier	0.2
K-Neighbors	0.28
Multinomial Naive Bayes	0.02
Decision Trees	0.16

5 Conclusions

5.1 Discussion

In this thesis we have proposed a new set of justification classes that can be used as a means of explanation for fact checking algorithms. Throughout our experiments we have proven that these new classes are more abstract than existing ones, meaning that they contain more explanatory information, and they are clearly understood by humans.
We have augmented the already existing *LIAR-PLUS* dataset, by assigning each claim-justification pair to one of our justification classes. This has been done through a human annotation process. Thus we publish this new dataset of our creation.
We have also proposed a method for evaluating the efficiency of the means of explainability that different fact checking methods use. We achieved this by creating a pipeline for each method that classifies the old means of justification to our new justification classes. So we have achieved evaluation by classification.
For the experimentation of the evaluation by classification method we have used two machine learning, fact checking, frameworks that provide explanations for their results in different ways. One provides a set of words and the other a short paragraph.
The result of the experimentation phase have shown that it is possible to map pairs of claims and justifications to more abstract classes with statistical significance, and hence our method can provide results.
Concerning the performance of our method we have observed the following

- The classifiers results, are placed near the mean of the defined range of values which is [*randomness*, *inter-annotator agreement*].
- We have found statistically significant results only for one of our test classifiers.

Based on these observations we can conclude that there is still room for improvement for our method. Our proposed improvements are show in the next section.

5.2 Future Work

During this research it became clear that there are some factors that can enhance the quality of our process but couldn't be part of this thesis. So for future work we propose the following.

5.2.1 Data Enhancement

The justification dataset, which is a crucial part of our research is very limited in terms of size, and this has played a major role on the classifier results. In order to improve this more data should be collected and annotated. An increase in the multitude of annotators should also be done in order to make the final dataset more robust.

5.2.2 Classifier Improvements

As shown in section 3.2.5 we have experimented mainly with traditional machine learning algorithms concerning the justification classifiers. We think that the use of simple deep learning models could enhance our results.

References

[1] Andreas Vlachos and Sebastian Riedel. “Fact checking: Task definition and dataset construction”. In: *Proceedings of the ACL 2014 workshop on language technologies and computational social science*. 2014, pp. 18–22.

[2] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. “Twitter under crisis: Can we trust what we RT?” In: *Proceedings of the first workshop on social media analytics*. 2010, pp. 71–79.

[3] Jacob Ratkiewicz et al. “Detecting and tracking political abuse in social media”. In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 5. 1. 2011.

[4] William Yang Wang. ““Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2017, pp. 422–426.

[5] Terry Lee. “The global rise of “fake news” and the threat to democratic elections in the USA”. In: *Public Administration and Policy* (2019).

[6] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding deep neural networks”. In: *Digital Signal Processing* 73 (2018), pp. 1–15.

[7] Sarthak Jain and Byron C Wallace. “Attention is not Explanation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 3543–3556.

[8] Sarah Wiegreffe and Yuval Pinter. “Attention is not not Explanation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 11–20.

[9] Kashyap Popat et al. “DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 22–32.

[10] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models”. In: *arXiv preprint arXiv:1708.08296* (2017).

[11] Pepa Atanasova et al. “Generating Fact Checking Explanations”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 7352–7364.

[12] Arthur L Samuel. “Some studies in machine learning using the game of checkers. II—recent progress”. In: *Computer Games I* (1988), pp. 366–400.

[13] Tom M Mitchell. “Does machine learning really work?” In: *AI magazine* 18.3 (1997), pp. 11–11.

[14] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[15] Elizabeth D Liddy. “Natural language processing”. In: (2001).

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[17] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[18] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. “Explainable artificial intelligence: A survey”. In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE. 2018, pp. 0210–0215.

[19] Anneleen Van Assche and Hendrik Blockeel. “Seeing the forest through the trees: Learning a comprehensible model from an ensemble”. In: *European Conference on machine learning*. Springer. 2007, pp. 418–429.

[20] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[21] Dong Huk Park et al. “Attentive Explanations: Justifying Decisions and Pointing to the Evidence”. In: (2017).

[22] Neema Kotonya and Francesca Toni. “Explainable Automated Fact-Checking: A Survey”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 5430–5443.

[23] Fan Yang et al. “Xfake: Explainable fake news detector with visualizations”. In: *The World Wide Web Conference*. 2019, pp. 3600–3604.

[24] Mohamed H Gad-Elrab et al. “Exfakt: A framework for explaining facts over knowledge graphs and text”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019, pp. 87–95.

[25] Naser Ahmadi et al. “Explainable Fact Checking with Probabilistic Answer Set Programming”. In: *Conference on Truth and Trust Online*. 2019.

[26] Sofia Serrano and Noah A Smith. “Is Attention Interpretable?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2931–2951.

[27] Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. “Where is Your Evidence: Improving Fact-checking by Justification Modeling”. In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 85–90. DOI: 10.18653/v1/W18-5513. URL: https://aclanthology.org/W18-5513.

[28] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. “Bidirectional LSTM networks for improved phoneme classification and recognition”. In: *International conference on artificial neural networks*. Springer. 2005, pp. 799–804.

[29] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: (2019).

[30] Cristian Bucilu□, Rich Caruana, and Alexandru Niculescu-Mizil. “Model compression”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 535–541.

[31] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* 2.7 (2015).

[32] Zhen Hai et al. “Deceptive review spam detection via exploiting task relatedness and unlabeled data”. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. 2016, pp. 1817–1826.

[33] Verónica Pérez-Rosas and Rada Mihalcea. “Experiments in open domain deception detection”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 1120–1125.

[34] Robert A Harris et al. “A handbook of rhetorical devices”. In: (1997).

[35] Maya Khemlani David. “Language, power and manipulation: The use of rhetoric in maintaining political influence”. In: *Frontiers of Language and Teaching* 5.1 (2014), pp. 164–170.

[36] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[37] Frank Wilcoxon. “Individual comparisons by ranking methods”. In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.