ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://eee.uniwa.gr
http://idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING
http://eee.uniwa.gr
http://idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Master of Science Thesis

Applying Graph Neural Networks for Stance Detection over Online Social Networks

Student: Monachelis, Panagiotis
Registration Number: AIDL-0009

MSc Thesis Supervisor:

Dr. Kasnesis, Panagiotis

ATHENS-EGALEO, February 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://eee.uniwa.gr
http://idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING
http://eee.uniwa.gr
http://idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Μεταπτυχιακή Διπλωματική Εργασία

Εφαρμογή Νευρωνικών Δικτύων Γράφων για την Ανάλυση Στάσης σε Μέσα Κοινωνικής Δικτύωσης

Φοιτητής: Παναγιώτης Μοναχέλης
Α.Μ.: AIDL-0009

Επιβλέπων Καθηγητής
Δρ. Παναγιώτης Κασνέσης

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Φεβρουάριος 2024

This MSc Thesis has been accepted, evaluated and graded by the following committee:

Supervisor	Member	Member
Panagiotis Kasnesis Digitally signed by Panagiotis Kasnesis Date: 2024.04.04 17:53:02 +03'00'	Maria Ragkousi Digitally signed by Maria Ragkousi Date: 2024.04.02 17:14:26 +03'00'	Digitally signed by Michail Papoutsidakis Date: 2024.04.05 14:02:44 +03'00'
Kasnesis, Panagiotis	Rangoussi, Maria	Papoutsidakis, Michael
Lecturer	Professor	Professor
Dept. of Electrical and Electronics Engineering	Dept. of Electrical and Electronics Engineering	Dept. of Industrial Design and Production Engineering
University of West Attica	University of West Attica	University of West Attica

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Παναγιώτης Μοναχέλης του Βασιλείου, με αριθμό μητρώου AIDL-0009 μεταπτυχιακός φοιτητής του ΔΠΜΣ «Τεχνητή Νοημοσύνη και Βαθιά Μάθηση» του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών και του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Η εργασία δεν έχει κατατεθεί στο πλαίσιο των απαιτήσεων για τη λήψη άλλου τίτλου σπουδών ή επαγγελματικής πιστοποίησης πλην του παρόντος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι 15 Μαρτίου 2024 και έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση του επιβλέποντος καθηγητή»

Ο Δηλών

Παναγιώτης Μοναχέλης

Declaration of the author of this MSc thesis

I, Panagiotis Monachelis Vassilios, with the following student registration number: AIDL-0009, postgraduate student of the MSc programme in "Artificial Intelligence and Deep Learning", which is organized by the Department of Electrical and Electronic Engineering and the Department of Industrial Design and Production Engineering of the Faculty of Engineering of the University of West Attica, hereby declare that:

I am the author of this MSc thesis and any help I may have received is clearly mentioned in the thesis. Additionally, all the sources I have used (e.g., to extract data, ideas, words or phrases) are cited with full reference to the corresponding authors, the publishing house or the journal; this also applies to the Internet sources that I have used. I also confirm that I have personally written this thesis and the intellectual property rights belong to myself and to the University of West Attica. This work has not been submitted for any other degree or professional qualification except as specified in it.

Any violations of my academic responsibilities, as stated above, constitutes substantial reason for the cancellation of the conferred MSc degree.

I wish to deny access to the full text of my MSc thesis until the 15th of March, 2024 following my application to the Library of UNIWA and the approval from my supervisor.

The author
Panagiotis Monachelis

Abstract

Fake news has flooded social networks and has been studied from different perspectives. Different methodologies have been proposed to approach this issue. In recent years, researchers have been conducting analyses based on machine learning and neural network models. The results of a model depend not only on the structure of the model by itself, but also on the right pre-processing of the data. For this reason, various techniques have been developed. The concern of researchers is to achieve the best results through the combination of the appropriate processing of the data with the appropriate model. The current MSc thesis examines the techniques developed in the field of assessing news based on rumors in social networks; to this end, the thesis implements an approach based on the Graph Neural Network algorithm (GNN). In the introductory part of the thesis, the techniques that have already been proposed by researchers are examined. In the next section, natural language processing operations are described with reference to embeddings that play a decisive role in transforming data into a format suitable for input into processing models. In particular, the thesis is based on data set analysis through Graph Convolutional Networks (GCN). For this reason, a detailed description of this model is provided; algorithms of the same category as GNNs are also presented. The data set used for analysis is described; this is a data set built specifically for research on the issue of rumor evaluation. The methodology adopted and employed is subsequently described. In particular, the open access data set that has been mined from the social network Twitter, contains rumors that have concerned the public and labels regarding the stance of users towards these rumors. The data set has already been analyzed through various methodologies, yet, not from the perspective of GNNs. The present thesis aims to analyze the same data set using the GNN architecture and comparatively evaluate results with existing, published results, in order to draw conclusions as well as identify limitations and provide feedback for further analysis.

Keywords

Social networks, Graph Neural Networks, LSTM, Fake news, Stance detection, Data veracity

Περίληψη

Οι ψευδείς ειδήσεις έχουν κατακλύσει τα κοινωνικά δίκτυα και έχουν μελετηθεί από διαφορετικές οπτικές γωνίες. Έχουν προταθεί διάφορες μεθοδολογίες για την προσέγγιση αυτού του θέματος. Τα τελευταία χρόνια οι ερευνητές διεξάγουν αναλύσεις βασισμένες σε μοντέλα μηχανικής μάθησης και νευρωνικών δικτύων. Τα αποτελέσματα ενός μοντέλου δεν εξαρτώνται μόνο από τη δομή του μοντέλου αυτού καθ' εαυτού, αλλά και από την σωστή προ-επεξεργασία των δεδομένων. Για αυτό το λόγο έχουν επίσης αναπτυχθεί διάφορες τεχνικές προ-επεξεργασίας. Κύριος στόχος των ερευνητών είναι ο συνδυασμός της κατάλληλης επεξεργασίας των δεδομένων με τη χρήση του κατάλληλου μοντέλου για να προκύψουν βέλτιστα αποτελέσματα. Η συγκεκριμένη διπλωματική εργασία εξετάζει τις τεχνικές που έχουν αναπτυχθεί στον τομέα της αναγνώρισης των ψευδών ειδήσεων που διαδίδονται μέσα από τα κοινωνικά δίκτυα. Για το σκοπό αυτό υλοποιεί μια προσέγγιση που βασίζεται σε αλγόριθμο Νευρωνικού Δικτύου Γράφου (GNN). Στο εισαγωγικό μέρος της διπλωματικής εργασίας αναφέρονται οι τεχνικές που έχουν ήδη προταθεί και δημοσιευθεί από άλλους ερευνητές. Στην επόμενη ενότητα περιγράφονται οι λειτουργίες επεξεργασίας φυσικής γλώσσας με αναφορά στις ενσωματώσεις (embeddings) που παίζουν καθοριστικό ρόλο στη μετατροπή των δεδομένων σε μορφή κατάλληλη για εισαγωγή σε μοντέλα επεξεργασίας, καθώς και σε μοντέλα που χρησιμοποιούνται στην παρούσα ανάλυση. Καθώς η διατριβή βασίζεται στην ανάλυση μέσω GNNs, γίνεται λεπτομερής περιγραφή τους. Το σύνολο δεδομένων που χρησιμοποιήθηκε για την παρούσα ανάλυση έχει αναπτυχθεί επί τούτου για έρευνα πάνω στο θέμα της αξιολόγησης φημών. Μετά το data set, στη συνέχεια περιγράφεται η μεθοδολογία που επιλέχθηκε και υιοθετήθηκε. Συγκεκριμένα, το σύνολο δεδομένων ανοικτής πρόσβασης που έχει εξορυχθεί από το κοινωνικό δίκτυο Twitter, περιέχει φήμες που έχουν απασχολήσει έντονα το ευρύ κοινό σε προηγούμενο χρονικό διάστημα και περιέχει επισημάνσεις (χαρακτηρισμούς) ως προς τη στάση των χρηστών απέναντι σε αυτές τις φήμες – πιθανές ψευδείς ειδήσεις. Το σύνολο δεδομένων έχει ήδη διερευνηθεί με διάφορες μεθοδολογίες, αλλά όχι από την οπτική γωνία των GNNs. Η παρούσα διπλωματική εργασία έχει ως στόχο να αναλύσει τα δεδομένα αυτά με τη χρήση της αρχιτεκτονικής των GNNs και να αξιολογήσει συγκριτικά τα αποτελέσματα ως προς υπάρχοντα αντίστοιχα αποτελέσματα μέσω άλλων προσεγγίσεων, προκειμένου να εξάγει συμπεράσματα αλλά και να εντοπίσει τους περιορισμούς και να παράσχει ανατροφοδότηση για περαιτέρω ανάλυση.

Λέξεις – κλειδιά

Κοινωνικά δίκτυα, Νευρωνικά Δίκτυα Γράφων (GNN), LSTM, Ψευδείς ειδήσεις, Ανίχνευση στάσης, Εγκυρότητα δεδομένων

Table of Contents

List of Tables

List of figures

Acronym Index

AI: Artificial Intelligence

BERT: Bidirectional Encoder Representations

CNN: Convolutional Neural Networks

CRF: Conditional Random Fields

GAT: Graph Attention Network

GCN: Graph Convolutional Network

GNN: Graph Neural Network

QA: Question-Answering

LLM: Large Language Model

LR: Logistic Regression

LSTM: Long-Short Term Memory

NBS: Naïve Bayes Classifier

NLP: Natural Language Processing

OSN: Online Social Network

RFC: Random Forest Classifier

RNN: Recurrent Neural Network

SVM: Support Vector Machine

TF-IDF: Term Frequency-Inverted Document Frequency

INTRODUCTION

The subject of this thesis

As social networks are now part of everyday life, the data circulating in them is voluminous and keep growing. Researchers study them to examine user behaviour from many perspectives, including the analysis of stance detection. Specifically, given an initial posting or twit in a social network, with a piece of news or with the expression of a specific point of view, users may respond to it by expressing different stances towards it: they might agree, disagree, question it or simply position themselves neutrally towards it. The stance assumed by users towards this piece of news is important if correlated with the veracity of the news: it should be alarming if the majority of the public would tend to agree or believe fake news. This thesis is about examining stance detection through Graph Neural Network. GNNs are neural networks proposed and developed more recently than other types of neural networks and have been used for many applications such as online social network analysis. The current analysis is based on a data set that is publicly available and built specifically for stance detection investigation, from data extracted from twitter. It is created on the basis of instances of various rumours that have been spread in the recent past regarding issues or event with considerable impact on the public; it is therefore interesting to examine the public viewpoint on them.

Background

The background for the current thesis consists of techniques that have been developed for analysing data from social networks as well as studies that have been conducted on the specific dataset. Concepts such as that of natural language processing and embeddings which involve pre-processing of text are presented and discussed; models used for stance detection, such as LSTM, BERT, and GNNs are presented. Previous existing studies using the same data set are also presented to highlight their effectiveness and to compare the results of this thesis with them.

Methodology

The methodology adopted and implemented consists of the following steps:

1. Pre-processing of the data set, such as cleaning and removing non-useful content.
2. Transformation of textual content to vectors-embeddings.
3. Creation of the graphs from the structure given by the data set.
4. Development of the models and analysis of the data set through these models. A GCN model and a combination of GCN with LSTM are used for analysis.
5. Training and evaluation of the models; results.

Innovation

The dataset has been explored by 8 different studies, each using a different approach. Approaches include

(i) *machine learning algorithms* such as linear regression and support vector machine and
(ii) *deep learning algorithms* such as LSTM and CNN.

Existing research publications do not include any GNN approach, however. The purpose of this thesis, therefore, is to implement the GNN approach and compare results with those of previous studies aiming to evaluate the performance of GNN on a rather standard yet not trivial data set.

Structure

The thesis in structured into chapters aiming to be easily readable and to make the content more understandable.

In the 1st chapter, the thesis presents a survey on existing research studies on social networks, the phenomenon of fake news and the methods that researchers have used to approach the issue of fake news. In contrast, the topic of the current thesis, i.e., stance detection, is highlighted.

In the 2nd chapter, the concepts of natural language processing (NLP) and of embeddings are described, focusing on the specific technique that has been used in the thesis. Also, popular neural network techniques used for this type of analysis are presented.

In the 3rd chapter, the concept of graphs and of graph neural networks (GNNs) are extensively described, focusing in particular on the most popular GNNs.

The 4th chapter reports on relevant work that has already been published on stance detection and specifically on studies that have used this particular dataset. The 5th chapter describes the implementation of the model proposed in this thesis, starting from the analysis of the dataset, continuing to the implementation of the graph neural network models and concluding with the results.

Finally, the 6th chapter concludes the study and raises issues for further consideration.

1 CHAPTER 1: FAKE NEWS AND TECHNIQUES OF ANALYSIS

1.1 Fake News on Social Media

Online Social Networks have become part of our everyday life and the amount of information that is circulated daily is enormous. According to the Eurobarometer 'Media & News Survey 2022' [1], 26% of EU citizens choose social networks as their media of choice, while according to Eurobarometer 516 [2], 55% of users up to the age of 24 choose social networks even for scientific information, compared to an EU27 average of 29%. The evaluation of information coming from social networks is more than necessary as it is clear from the above data that the information circulated on social networks has a strong influence on users.

Social media have the advantages of easy access and rapid spreading of information. On the other hand, they have the disadvantage of spreading low quality news and sometimes intentionally false information that may have negative impact on users individually or on the society in general. It is remarkable that during the election period in the United States of America, in 2016, the average American adult saw one or more fake news stories in the social media and more than half have believed them [3]. The politics domain is just one of the domains where fake news have a strong influence. The issue of misinformation is getting more crucial as it may put at risk the society in cases where antiscientific news is spread, endangering people who tend to believe conspiracy theories. This fact became particularly noticeable during the recent pandemic of COVID-19. It is indicative that the topic of COVID-19 has been investigated more than any other in the context of conspiracy theories and in general, public health topics are at the top of the list with high publication counts; furthermore, among them numerous are the studies on social media use, conspiracy beliefs and health-protective behaviours [4]. This situation raises a concern about misinformation and its impact, from the perspective of trust in science. The importance of 'fighting' fake news on social media lies in the fact that social media are the platforms where fake news is most widely circulated, such as in the case of COVID-19 according to the study of Naeem et al. [5]

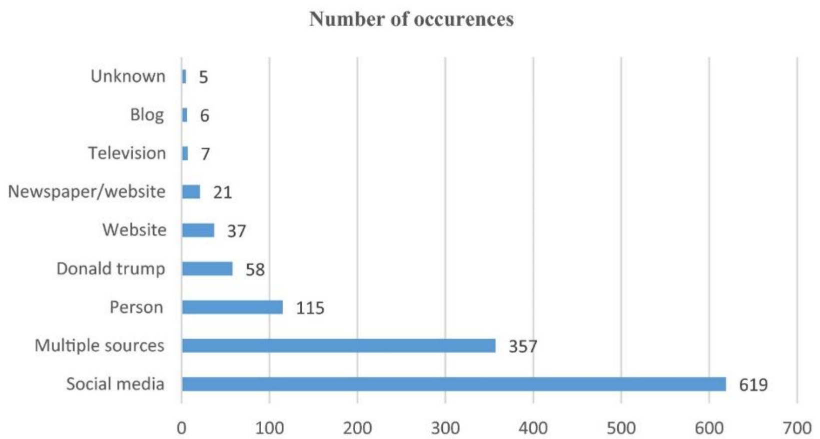

Figure 1: Ranking of media related to fake news dissemination during Covid-19 pandemic [5]

The figure 1 depicts the dissemination of fake news in different media. It is impressive that misinformation takes place in social media much more frequently than in other sources. Consequently, fake news detection in social media is a task of a high interest and attracts the attention of researchers to face this challenge. It is a demanding process because the mere content of a piece of news is often not adequate for the determination of its validity. For the best possible identification of fake news, [6], auxiliary information must be considered, such as social engagements of the user on social media. This auxiliary information is related to specific features:

- The source of the news, which can be an author or a publisher.
- The headline that is a short title describing the main topic and aiming to attract the attention of readers.
- The body text that contains the detailed view of the author.
- Media files of images or video that accompany the news to provide an extra visual content.

In addition to investigating the characteristics of a news and posts in social media, the analysis can be conducted according to the perspective from which it is viewed. There are different thematic areas which approach the fake news detection from different perspectives.

Rumour detection: Rumour is the ambiguous information that is circulated and has not been verified. Analysis from this perspective has been conducted in order to examining the validity of rumours. Approaches that have been proposed include rumour detection, rumour tracking, stance classification and veracity classification, [7].

Truth Discovery: It refers to the investigation of multiple sources containing possibly opposing views, aiming to determine the truth. A crucial strategy for this task is the investigation of source credibility [8].

Clickbait detection: 'Clickbait' headlines aim to attract reader attention causing readers to click on the link of the article and thus result in advertising revenue for the host. The approaches for clickbait detection use linguistic features such as the similarity of the title with the main content, as sometimes the clickbait is misleading, or informality, as clickbait headlines tend to use less formal language than professional articles [9].

Spammer and Bot detection: Fake news can be spread by bots automatically; it is critical to detect such sources, as they can spread misinformation fast and widely. Bot identification research involves, among other approaches, linguistic analysis that examines the features of the bot content, especially part-of-speech (POS) tagging, such as the frequency of verbs, nouns, and adverbs in the messages [10]. Fake content also can be created by spammers such as opinion spams; linguistic approaches have been used for their detection [11].

1.2 Techniques and Approaches on Fake News Detection

According to the above, fake news detection is a demanding task and concerns the researchers from different perspectives. The linguistic approaches include various methodologies. As there is a big amount of data that has to be processed, the selected techniques should employ effective algorithms that can handle the size and the complexity of these data. Various types of analyses have been conducted to address misinformation through different methodologies; major ones are listed below.

A first categorization of the different techniques used for this task that employ machine learning models, is into

(i) Non-Neural Network models and

(ii) Neural Network models.

Machine learning methods that have been used frequently are the Support Vector Machine (SVM) and Naïve Bayes Classifier (NBC), while Logistic Regression (LR) and Random Forest Classifier (RFC) are used occasionally [10].

From the perspective of Neural Networks, the Recurrent Neural Network (RNN) is very popular and especially the Long-Short Term Memory (LSTM) model. Convolutional Neural Networks (CNN) are also widely used, while combinational models of CNN and LSTM have also been developed, as attention mechanisms of LSTM are incorporated into models for better performance.

A study was conducted by Aldwairi and Alwahedi [13], aiming to identify a solution to detect and filter out websites that contain fake news. The researchers have developed a tool that identifies and removes sites with fake news from the results of a search engine or of a news feed of a social media platform. As a first step, they collected URLs from social platforms as there it was more likely to find fake news; as a second step, they gathered information about the title and the content of each article from the website. They used machine learning classifiers such as Bayes Network, Logistic, Random Tree and Naïve Bayes. The results of these classifiers were compared as to the Precision, Recall, F-Measure and ROC values, with the Logistic classifier yielding the highest precision (99.4%).

A survey on fake news detection identified a plethora of SVM models as well as other machine learning techniques that have been used for this task, yet less often, such as Conditional Random Field (CRF), Decision Trees (DT) and Hidden Markov Models (HMMs) [14]. Although traditional machine learning algorithms seem to have superior performance, they have a disadvantage over neural network approaches as their representations are based on manually calculated features; this makes model development time-consuming and may result in biased features, too. Also, a comparative study between traditional machine learning algorithms and neural networks on a text classification task showed that CNN architecture had superior performance compared to five different machine learning models [15].

Apart of technical implementation and the right choice of a model, there are different approaches of detecting fake news and assessing their impact. Beyond the statistical investigation involving the counting of indicators such as 'likes' and shares of posts in order to find high-impact messages, artificial intelligence techniques analyse messages by drawing conclusions such as the sentiment expressed through a message or the objectivity/subjectivity of a message. Researchers aim to derive results based on the analysis of the features of the content, such as sentiment analysis or stance detection. As the authors/creators of fake news often create the misleading information aiming exactly to excite the sentiment of the readers, sentiment analysis is one of the tasks that frequently accompanies the detection of fake news [16]. Another major task in the process of fake news detection is stance detection. The analysis in study [16] also extracts results about the opinions of people toward fake news articles aiming to identify if the news has impact on its audience or not.

1.3 Approach of the current thesis

The major task undertaken in the current thesis is stance detection, i.e. the detection of the stance expressed by a social network user in response to / towards an initial message. This analysis studies a data set in which users agree, disagree, quest or have a neutral stance towards a previous message. Stance detection can be applied simultaneously with other methodologies such as sentiment and subjectivity analysis for a more in-depth research, to indicate whether a message contains positive, negative or neutral sentiment, whether a message agrees, disagrees or remains neutral towards another message and whether it expresses an objective or subjective viewpoint. The above techniques fall under the general category of Natural Language Processing (NLP) methods. For stance detection, the current thesis uses a dataset deliberately created for this reason by its authors, i.e., built in order to study stance detection models on it. Furthermore, the thesis employs a neural model, namely, a Graph Neural Network model, to analyze the data set and to comparatively evaluate its effectiveness on the given data set.

2 CHAPTER 2: BACKGROUND

2.1 Natural Language Processing

As social media are flooding the Internet, the volume of user information is becoming increasingly large. This circulated information is mainly in text format; to automate its analysis, it needs to be transformed in a format readable by (computing) machines. Also, processing and conclusion drawing from all this data requires a high degree of automation, i.e., the use of rapid and efficient methods, which is not possible by humans. NLP performed by computers is necessary in order to manage this massive text data. NLP is based on a multitude of methods and is used in many applications where natural language translation leads to the development of very useful tools, such as

- Text Classification in categories,
- Information Extraction (IE),
- Question-Answering (QA),
- Automatic Language Translation,
- Opinion Mining,
- Sentiment Analysis,
- Stance detection,

and others [17].

However, the development of a model for understanding natural language is a great challenge, given that languages represent large and complex sets of data (words, vocabularies, etc.). There are additional difficulties in machine understanding of human languages, due to, e.g., the double meaning of certain words. The difficulty lies not only in the large number of words and their semantics, but also in understanding the syntax of sentences since each language has its own peculiarities. Machine learning comes to provide efficient solutions to this challenge of developing effective methods for computer understanding of natural language.

The NLP procedures include steps of pre-processing such as

- Tokenization,
- Part of Speech (POS) tagging,
- Word segmentation, and
- Parsing.

In Tokenization, a period of text or a longer document is split into ‘tokens’ based on the spaces existing between the words. A next step is stop-word removal, where short-lettered, frequent, yet unimportant words, named ‘stop-words’ are removed (“the”, “a”, “not”, “will”, pronouns, etc.). On the other hand, segmentation is a process where a word is divided into its syllables. This is more necessary in languages such as Chinese or Japanese in which there do not exist clear word boundary markers. POS tagging and parsing analyse respectively the lexical and syntactic information of a period of speech or text, resulting in tags such as ‘subject’, ‘verb’, 'object', etc. Different tools have been developed to accomplish these NLP tasks, such as NTLK and Gensim in Python, or OpenNLP and CoreNLP in Java [18].

NLP-related services are also provided as integrated solutions for application development by platforms such as Amazon and Microsoft. The tools include a variety of services such as language detection, sentiment analysis, question answering, speech and text translation and conversational AI [19].

2.2 Embeddings

Embeddings constitute an important component for NLP, as they transform text to formats that can be understandable by machines. Language models accept inputs such as vectors, that can be processed by processing units or subjected to mathematical operations. Different models have been proposed for the representation of words or periods by embeddings, Word2Vec and Glove are the two models mostly used today. Their success lies in the fact that they are efficient, as they learn high-quality word embeddings from very large corpora and at the same time have a user-friendly implementation [20].

Embeddings can be categorized in three categories (Figure 2), [21]:

1. *Traditional Word Embeddings*,
2. *Static Word Embeddings* and
3. *Contextualized Word Embeddings*.

Traditional word embeddings take in account the frequency of occurrence of the words in the whole document, as well as the co-occurrence of words in the whole body of texts and thus reveal the significance of rare words. The most used approaches of this category include the techniques of

- Count Vector based on a basic descriptive statistic of the words,
- Term Frequency-Inverted Document Frequency (TF-IDF) giving information of how often a term is appeared in a document but also providing information about the rarity of a term, and
- representation of co-occurrence of words in Co-occurrence Matrix [21].

Static Word Embeddings are a probability approach and represent words to vectors. Such an embedding is characterized as static because it does not change the vectors of words after the model has learnt and produced the vectors. The major models of this categories are:

1. Word2Vec: it is a model that analyses the semantic similarity between words,
2. Glove: it has a similar concept but its implementation differs as it employs a co-occurrence probability ratio,
3. FastText: it is a family of techniques with a wide use in cases such as the analysis of semantic similarity, and
4. FastText that uses the Skip-Gram model (used in Word2Vec as well): they investigate word representation considering subword information [22].

Contextualized Word Embeddings: in contrast to static embeddings, contextualized embedding can change the vectors of the words dynamically as they are based on the context of a particular word, that may assume different meanings within different sentences. Approaches of this category are:

1. ELMo embedding that is pre-trained on a large text corpus and capture the syntactic and semantics information of word with different context,
2. BERT model that uses multilayer bidirectional transformer encoders for language representations and learns the contextual relations between the words or subwords handling sentences of minimum length, and
3. GPT 2 model, a decoder transformer only, that predicts words looking at parts of a sentence and it is trained over millions of websites [21].

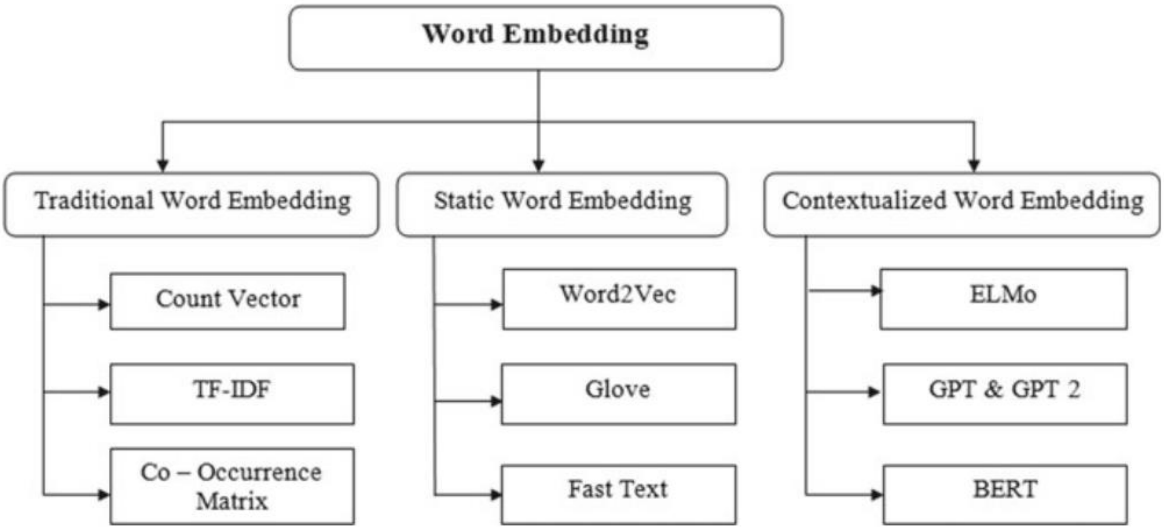

Figure 2: Types of Word Embedding techniques [21]

2.2.1 Word2Vec

Word2Vec is a model architecture for computing continuous vector representations of words that come from very large data sets and it was proposed by Google [23]. Simpler models follow a frequency-based approach; in contrast, Word2Vec captures and exploits syntactic and semantic word similarities. The researchers of Google had proposed two architectures, the continuous Bag-of-Word model and the continuous Skip-gram model. The first one predicts the current word based on the context while the second one predicts surrounding words given the current word. These models have much lower computational complexity compared to neural networks such as feedforward and recurrent networks, and have the ability to compute accurately high dimensional word vectors from a larger data set.

The operation of the Word2Vec is based on features such as the *window size*, which is a frame that captures the neighboring words around a specific word, that are used for the analysis of the context, and the *vector dimensions*. The importance of the Word2Vec is that can compute the similarities of the words from the training of a large corpus. Similarity is calculated using the Cosine Similarity equation while the value of the similarity ranges from -1 to 1, corresponding to the lowest and the highest similarity, respectively [24]. An example of similar words is depicted in the figure 3.

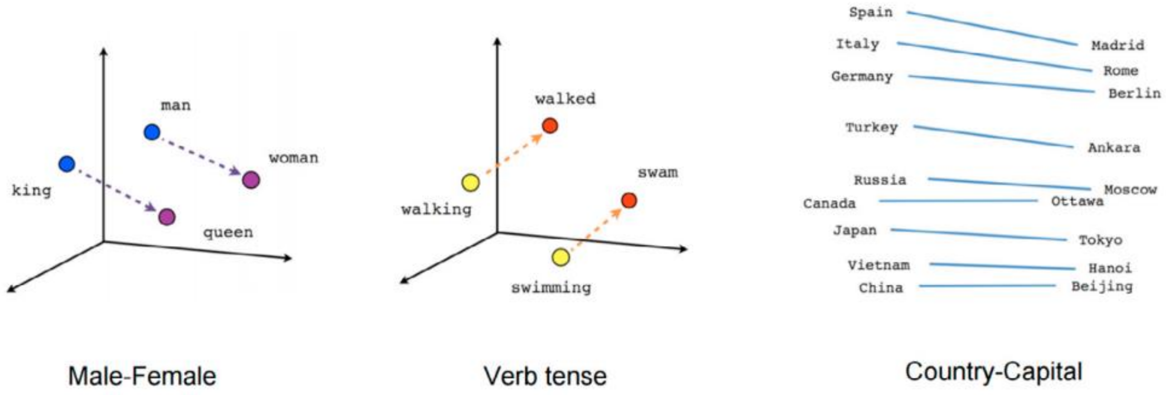

Figure 3: Word2Vec representation: examples of word similarities [24]

2.3 Long-Short Term Memory Model

LSTM model is the most commonly used RNN. As it is a part of the development of this thesis, it is deemed necessary to describe the model along with its features and applications. RNNs have been used for sequential data processing and analysis, such as text, audio and video. They can take in account previous inputs of the model but they have a short-term memory: only recent past is considered while the rest is progressively faded out and 'forgotten'. A disadvantage of RNN is that, given the practical limitation of memory size, when its memory runs out, previous information is deleted in order to capture the new information. LSTM model is a variation of RNN introduced to tackle exactly this situation. A schematic overview of the LSTM model is depicted in the following figure 4.

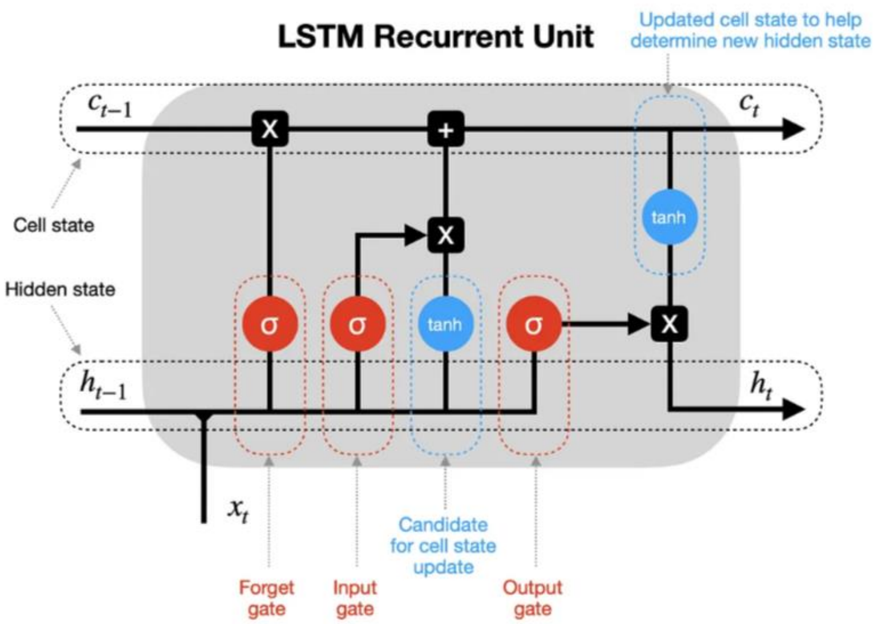

Figure 4: An LSTM cell: schematic representation [25]

According to figure 4, the structure of an LSTM cell includes a combination of sigmoid and tanh functions which form 3 different gates:

- the *forget* gate,
- the *input* gate, and
- the *output* gate.

These gates produce as output a number between 0 and 1 and are trained using the backpropagation algorithm. Also, the cell includes four different states: the cell state (c_t), the hidden state (h_t) and the cell and hidden state of the previous cell (c_{t-1}, h_{t-1}). In figure 4, x_t represents the current input. The information propagated on the straight line on the top part of the cell represents the cell state that progresses from c_{t-1} to c_t. The hidden state that characterizes the previous data, as depicted at the bottom line of the cell, also progresses from h_{t-1} to h_t. Information is added or removed to the cell state via the 3 gates. The gates are capable to process the data and decide what information is relevant in order to 'keep' or 'forget' the data:

- the forget gate decides what information to be kept or removed having as input the current input and the previous hidden state,
- the input gate updates the cell state while
- the output gates produce the hidden state and thus decides which information will be led to the output [26].

The LSTM model has found a variety of applications; their variety is indicative of its importance. For example, LSTM has been used to improve the services offered by Google for speech recognition and Google Translate machine translations; it has improved the answers of Alexa by Amazon; Facebook has used this model for translation services. The model has been successfully used for gaming (AlphaStar by Google DeepMind), financial market predictions that outperformed traditional benchmarks, to forecast oil market prices, to predict taxi services demand, for activity recognition based on data from wearable sensors, and many others [27].

2.4 Transformers

In 2017, Google released the Transformer model for the first time [28]. Other models, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) were already being used in NLP tasks by that time. The proposed model of the transformers, however, outperformed existing state-of-the-art NLP models, thanks to the advantage of transformer models to be trained faster and to achieve higher evaluation scores [29].

The transformer model is based on an architecture of encoder-decoder with multi-head self-attention. The initial model was trained on a data set of 4.5 million pairs of English-German sentences and of 36 million pairs of English-French sentences. The innovation of the transformer model over existing ones was the positional encoding that feeds the model not just the words but also their relevant positions in the sentence, along with the respective embeddings. The structure of words in a sentence is thus taken in account to improve results. Also, the self-attention mechanism incorporated in the transformer models relates the words in a sentence so that they can be understood by their context in the sentence.

The original architecture of the transformer model was the basis for variations subsequently developed, to solve more demanding transduction problems and improve language modelling.

2.5 Encoder-based Transformers

One of the most popular transformer-based models is BERT (Bidirectional Encoder Representations from Transformers). BERT was introduced by Google developers and exhibited superior performance in eleven (11) NLP tasks, including text classification, sentiment analysis, semantic role labelling, and the disambiguation of words with multiple meanings. BERT architecture is based on the original Transformer model architecture and it consists of a multi-layer bidirectional encoding-only Transformer. There are two BERT models:

- BERT-base, with 12 encoder layers and 110 million parameters, and
- BERT-large, with 24 encoder layers and 340 million parameters [30].

BERT has been designed to generate a language model and includes only the encoder part of a typical encoder-decoder Transformer model. The successful methodology that fuels BERT is Mask Language Model. This model masks in a random way the tokens in the input and then predicts the original words based on the surroundings (context, neighboring words). It thus achieves a better word representation as it considers both left-hand side and right-hind side parts of the sentence, around a specific word [31].

BERT is used for various NLP tasks:

i) classification tasks, such as sentiment analysis aiming to classify the text in categories of stance or attitude (e.g., positive, negative or neutral),
ii) question-answering models, where the model is trained to understand the question and give the appropriate answer,
iii) named entity recognition, aiming to predict the terms that represent a person or organisation in the text,
iv) conversational AI systems, where BERT is trained to function as a chatbot or virtual assistant, and
v) text representation, where BERT is used to generate the embeddings of a text.

Different variations of the initial BERT algorithm have been developed, such as [32]:

1. ALBERT: A Lite variation of BERT, designed with reduced parameters and lower memory consumption as compared to full BERT, yet, without loss of performance in pretraining large models.
2. RoBERTa: A BERT model trained using larger data sets and for a longer period.
3. DistilBERT: A distilled variation of BERT, for faster and efficient modeling that requires less computational power.

Additionally, there have been developed specific models that have been trained for more specific reasons, such as BERTweet, a model based on a large English Twitter dataset, pretrained with RoBERTa.

2.6 Graphs and their importance

A graph is a data structure that contains two components: Vertices (or nodes) and Edges (or links). The nodes are connected via links producing a graph. A node may represent different objects while links can represent interactions, connections or communication/interfacing between nodes. In order to obtain a formal description for a graph G, in general it is considered that any graph G is consisting of nodes V and edges E between the nodes, described by G = (V, E). A typical example of a graph is a social network of users: nodes may represent users and the existence of a link between two nodes (users) may indicate their connections as 'friends' in the specific social network (graph), [33]. As a graph structure may become arbitrarily dense, its representation is usually not possible on the Euclidean system / 2D plane. This feature is unlike other data structures, such as images, where each pixel has the same neighborhood structure and can be depicted on the 2D plane [34] (Fig. 5).

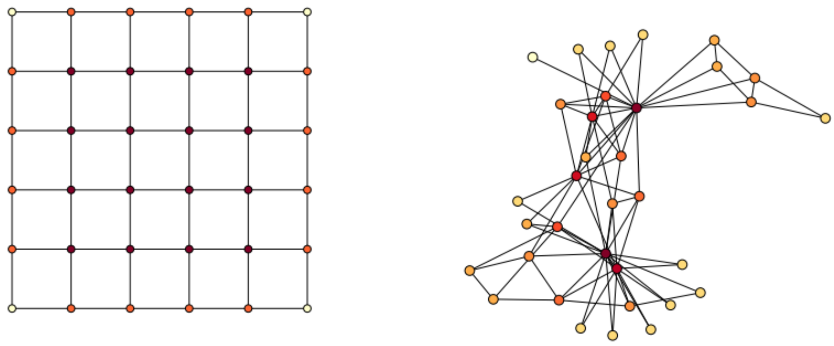

Figure 5: An illustration of a Euclidean graph (left) and a non-Euclidean graph (right) [34]

For this reason, graphs constitute a separate and more advanced data structure that has long been studied in the field of algorithms and complexity. There, a graph is described by an adjacency matrix (X_{ij}) [35]. The *ij*-th cell of the matrix, X_{ij}, is a value representing the relationship between the specific pair of two nodes (i, j). It is equal to 0 when there is no connection between the two nodes and may assume different non-zero values according to the problem addresses. For example, when the graph is of the binary type, the value of cell $X_{ij} =1$ indicates that nodes i and j are connected. Furthermore, when the problem addressed requires that the graph is directional, the matrix may not be symmetric, while for non-directional graphs the matrix is symmetric: $X_{ij} = X_{ji}$ for all (i, j).

According to Zhou et al. [36], who have carried out an extended review of methods and applications of Graph Neural Networks, graphs may support two kinds of scenarios:

- structural scenarios, where the graph is explicit, and
- non-structural scenarios, where the graphs are implicit and the scheme of the graph must be created.

The structural scenarios find applications in research problems involving molecules, knowledge graphs and physical systems, while the non-structural scenarios are used in applications such as building scene graphs for images or the creation of connected words in a text. In the same study, the types of graphs are categorized in three types:

- *Directional and Undirectional Graphs*: Nodes in directional graphs are connected with specific direction from one node to another, while nodes in the undirectional graphs have no directed links; in that case, a non-zero link indicates only that there is a connection of the 2 nodes without any directionality.
- *Homogeneous/Heterogeneous Graphs*: In a homogeneous graph, all nodes and edges are of the same type, while in a heterogeneous graph the nodes and the links may be of different types, according to the problem addressed.
- *Static/Dynamic Graphs*: The parameter of time plays a significant role for this distinction. When a graph changes over time then it is dynamic, while a graph with stable form over time is considered as static. Mathematically, the value of a cell in the graph matrix is a constant ($X_{ij} = c_{ij}$) with time for static graphs, while it becomes a function of time, $X_{ij}(t)$, for dynamic graphs.

2.7 The concept of a GNN

A literature survey has revealed that GNNs were first introduced as a graphical-based learning environment for pattern recognition [37]. The researchers proposed a neural network model suitable for both node- and graph-focused applications and they named it Graph Neural Network, as an extension of both random walk models and recursive neural networks. Subsequently, the main author of this first publication further evolved the architecture and published the paper “The Graph Neural Network Model” which was the most cited paper on this topic until the time of the literature survey.

The use of GNNs can be split into four main categories [38].

Node Classification: The goal of this type of machine learning graphs is to predict the label of a node, which could be a category, type, or attribute. A typical example of node classification method is the identification of the nodes in a social network, where a model could predict if a node belongs to a user or to a bot. Beyond social networks, GNNs can be applied in other domains as well, such as applied chemistry [39] and recommendation systems [40].

Relation Prediction, or link prediction or relational inference is the methodology that is used for predicting the relationship between the nodes and it can be applied in different areas such as social platforms, predicting drug side-effects [41] and inferring new facts in a relational database [42]. The model consists of nodes and an incomplete set of edges aiming to the prediction of the missing edges.

Clustering and Community Detection: The models of this category aim to identify similar features of nodes and edges trying to form groups with common features in a network. An example of community detection is the citation of research papers, where researchers reference each other in their articles. In a graph of interactions, the papers and the researchers are represented by nodes and the edges show the citation of each other. Dense subnetworks in the whole graph reveal the common thematic areas of researchers and their similar scientific field. Another application of clustering and community detection is in genetic interaction networks [43] and in financial transaction networks uncovering fraudulent groups [44].

Graph Classification: The methodology of Graph classification is based on feeding the models with multiple labelled graphs and instead of making predictions about nodes or edges individually as in the case of the above node and edge prediction models, the goal of a graph classification model is to make predictions at a graph level trying to interpret the features of the unlabeled graph. For example, a series of molecules graphs with known features such as toxicity or solubility can predict the properties of unknown molecules [45]. Also, another model has been implemented to predict the behavior of a computer program trying to identify it is malicious or not according to the structure of the syntax [46].

2.8 GNN models

There are different types of GNNs that a survey has concentrate in the following figure:

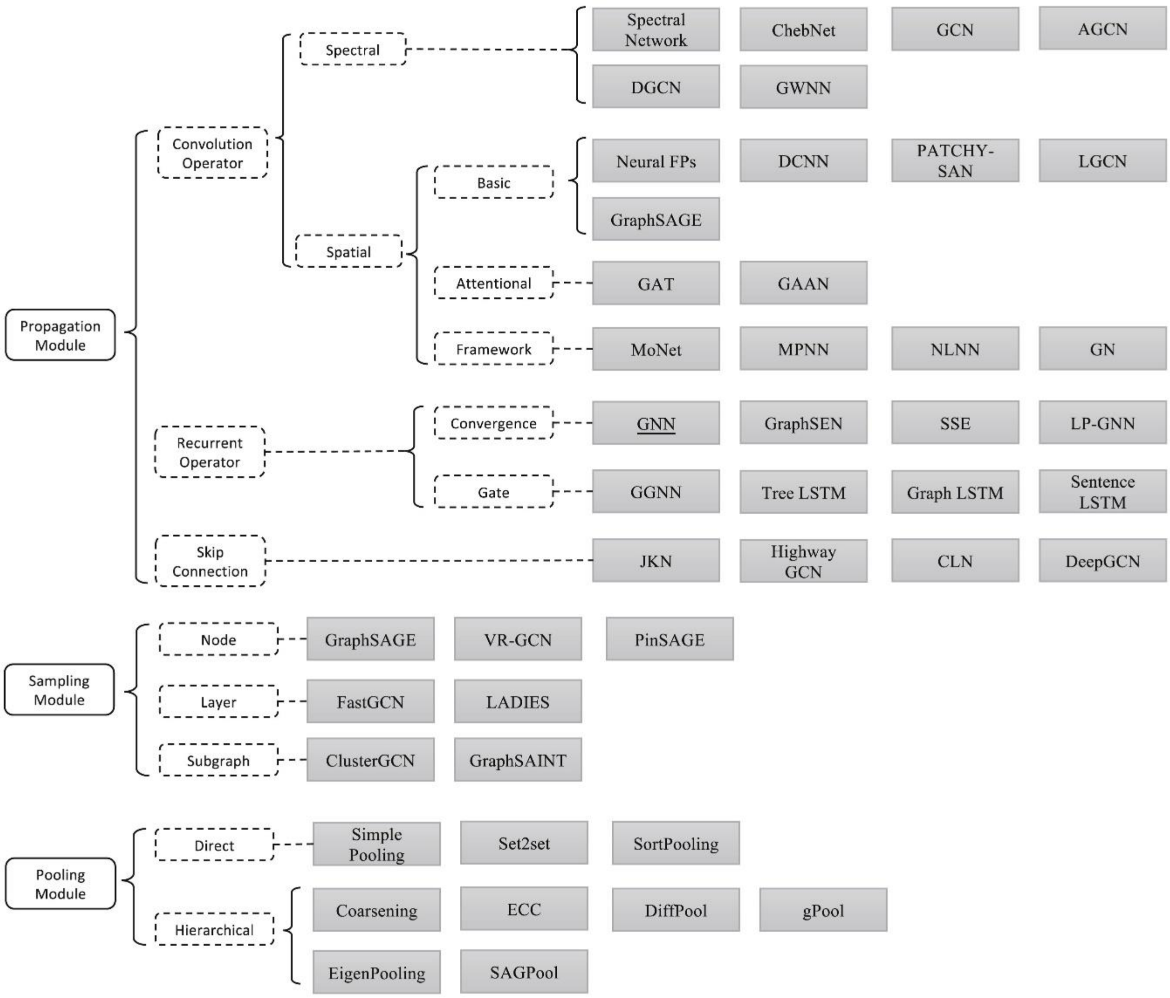

Figure 6: Computational methods of GNNs [36]

The figure 6 illustrates a general overview of the GNN models that are categorized in three main categories. Propagation modules with the majority of them to use convolution operator whose main idea is the adoption of convolutions from other domain to graphs. The models of convolution graphs are separated to spectral and spatial models. Sampling modules administrate the overload of information in a graph. Nodes in GNNs aggregate information of neighboring nodes in previous layers and multiple layers can overload the graph with extensive and redundant information and sampling modules are necessary to conduct the propagation. Also, pooling modules are referred to the pooling layer that is usually following a convolution layer. The most commonly used models will be described below.

2.8.1 GCN – Graph Convolutional Network

The GCN architecture was introduced by researchers from the University of Amsterdam [46]. It is a type of convolutional neural network that has brought significant performance on images; furthermore, it can be applied on graphs and has therefore the benefits of this data structure as well. The main idea of GCN is that, for each node in a graph, the node aggregates information of its neighboring nodes and applies a function on it, such as averaging, to estimate values that are representative of all nodes in the neighborhood; these values then feed the neural network input layer.

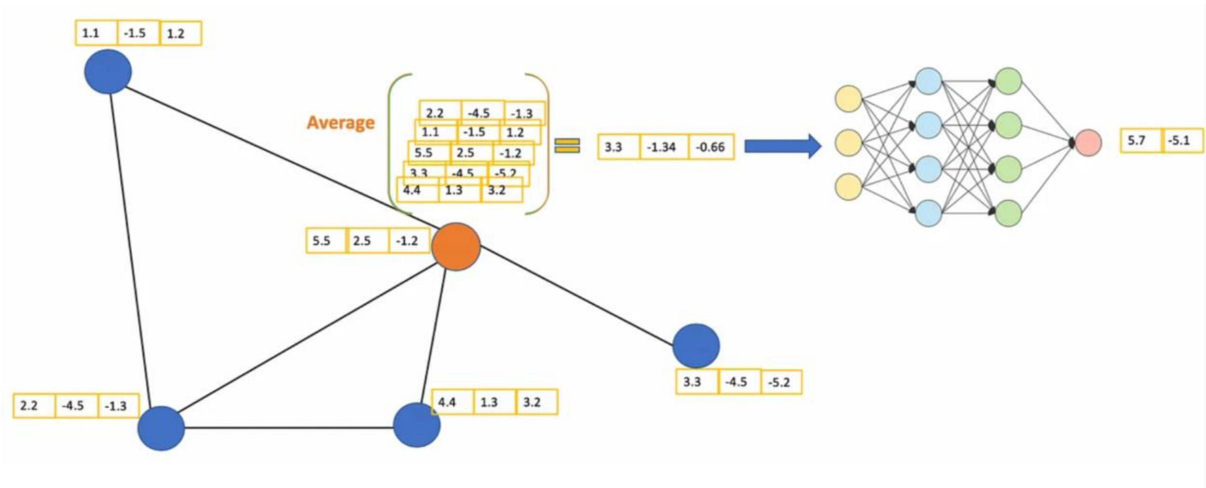

Figure 7: The operation of GCN [47]

In figure 7, the orange node aggregates the features of its four neighbors. After the computation of averages, a vector of three average values is produced which in turn will be fed to the neural network as input. All the graphs are fed to the input GCN layer, which in turn may feed a second GCN layer in order to produce the output through an activation function, such as Rectified Linear Unit (ReLU) (Fig. 8).

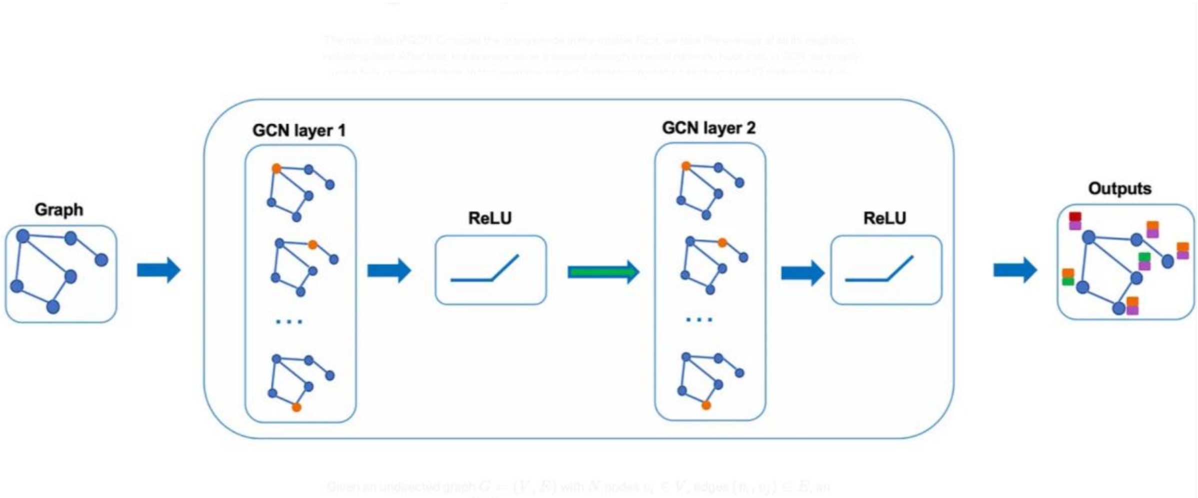

Figure 8: Convolutional layers in GCN following by ReLU activation function [47]

2.8.2 Graph Attention Network (GAT)

The GAT model was introduced in 2018 [48] as a novel convolution-style neural network model. It incorporates a self-attention mechanism that was useful in the cases of machine reading and learning sentence representations, in order to address the deficiencies of the convolutional network models. The main idea is the computation of the hidden representations of the nodes in a graph, taking information from its neighbors and following a self-attention methodology. The validation of GAT is proven by the results of the application of the model in four challenging benchmarks; there, GAT has achieved state-of-the-art results that prove the effectiveness of the attention-based models in the cases of arbitrarily structured graphs. The main difference between GAT and the GCN model is that GCN aggregates the information of the neighboring nodes relying on simple structural information such as node degree, while in contrast, GAT model aggregates the features from the neighboring nodes, such as embeddings. In this way, the weights of the model depend on more than just the count of neighboring nodes but may include any of the features of the nodes. These are calculated by the attention function, as this method allows some nodes to attend other nodes more than others. Computationally, the GAT model needs more resources compared to GCN, because it executes more calculations for the attention coefficients for each pair of connected nodes1.

2.8.3 GraphSAGE

This approach is a general inductive framework that generates node embeddings for unseen nodes based on the known neighboring data [48]. The novelty of this method is that embeddings of a node can be derived from the aggregation of neighboring nodes and it is not necessary to train for individual embeddings for each node.

[1] https://medium.com/@farzad.karami/understanding-graph-attention-networks-a-practical-exploration-cf033a8f3d9d

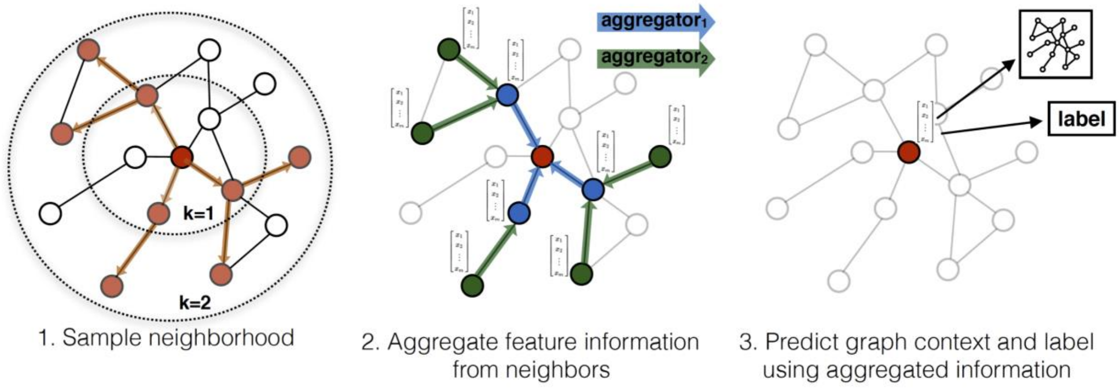

Figure 9: Sampling and aggregation of the GraphSAGE model [48]

In figure 9, the light red nodes are sampled and then the aggregation is conducted to provide the features of the central red node. The difference between GraphSAGE and GCN is that GCN is applied in the transductive learning with fixed graph while GraphSAGE is applied for an inductive unsupervised learning using trainable aggregation functions. The evaluation of the algorithm was based on its application to 3 node-classification benchmarks; results have shown that this method can produce effective representations for unseen nodes, as the model outperformed relevant baseline results.

3 RELATED WORK

3.1 Stance Detection

Researchers have investigated the data derived from OSNs from different perspectives. Among various other NLP tasks, a quite interesting one is stance detection. According to Biber and Finegan [50] "*By stance we mean the overt expression of an author's or speaker's attitudes, feelings, judgments, or commitment concerning the message*". In the case of OSNs, stance detection has been used to study the stance of the author of a post message as a reply to another message. In OSNs, a post can contain a message expressing a view; replies to this message may be supportive, opposing, questioning, or simply neutral. Stance detection is often confused with sentiment analysis, so it is noteworthy to mention their differences. Sentiment analysis is a NLP task used to determine the nature, strength and polarity of emotion(s) expressed by the author in a text: these emotions may be positive, negative or neutral; they may be strong or moderate; furthermore, there may be detected one or more of a spectrum of emotions of interest with regard to the application or problem at hand (anger, interest, boredom, enthusiasm, joy, satisfaction, stress, etc.) Stance detection, on the other hand, focuses on identifying a person's view toward an object of evaluation [51]. Therefore, stance detection is of particular importance for the study of the impact of the content circulated within social platforms.

For the investigation of stance detection models, various data sets have been developed / compiled [52] such as:

- RumourEval-2017: Tweets of rumours of 8 different topics have been collected to study if a tweet has the stance of support, deny, query or comment towards to another tweet.
- FNC-1: A dataset for fake news examination. Specifically, the task determines the stance of the news body text towards the news headline. The content of the body text is labelled as 'agree', 'disagree', 'discuss' or 'unrelated' to the headline.
- FEVER: A dataset that is used for fact-checking and contains sentences from Wikipedia that are classified as Supported, Refuted or NotEnoughInfo.
- RumourEval-2019: It contains tweets in a conversation thread towards the rumor mentioned in a source tweet. The tweets are labelled as support, deny, question (the reply requires more info for the veracity) and comment (not clear stance).
- COVIDLies: A stance detection dataset consisting of 6761 tweets; it provides annotation on COVID-19 misinformation content.

Different methods for stance detection have also been developed to determine the stance of a sentence in an automated way, using machine learning and deep learning algorithms.

According to Lai et al. [53], the first approach to stance detection was implemented by Somasundaran and Wiebe [54] who presented an unsupervised method for debate-side classification, trying to identify which stance a person takes towards an online debate. The topic of the content was online technological debates, such as between two phone models (iPhone and Blackberry) adopting an Integer Linear Programming approach.

Support Vector Machines (SVM) techniques have also been used to investigate the stance detection on tweets about two popular sport clubs in Turkey [55]. Two SVM classifiers were trained (one for each target-sport club) using unigrams as features in order to predict the content of the tweet as 'against' or 'in favor'. For training, 'hashtag' has been used as an extra feature; it is reported that the efficiency of the model was slightly better when including 'hashtag'. The use of additional features, such as the existence of external websites, or positive or negative emoticons, didn't lead to improved results.

A Convolutional Neural Network architecture has been used for the needs of stance detection in the SemEval2016 Task 6, [56]. The dataset contains tweets on six different topics (Hillary Clinton, Feminist movement, Legalization of Abortion, Atheism, Donald Trump, Climate change) that are manually annotated as 'favor', 'against' and 'neither'[2].

A comparison of sequential and non-sequential classifiers has been conducted testing a rumour dataset containing tweets associated with eight events in tree-structured conversations labelled with their stance[3]. Specifically, the sequential classifiers that were examined were Linear Conditional Random Fields (CRF), Tree CRF, and LSTM while the non-sequential classifiers used were SVM and Maximum Entropy (MaxEnt). In general, the sequential classifiers outperform non-sequential classifier and, interestingly, LSTM seemed to outperform all others, [57].

A combinational model of CNN and LSTM has been studied in the context of the Fake News Detection[4] contest. There it was proved that the use of combinations of CNN+LSTM or LSTM+CNN leads to better results than the use of CNN or LSTM alone, respectively, [58]. For the aforementioned contest, another approach that had superior performance was a combination of CNN+LSTM using the Principal Component Analysis (PCA) technique for dimensionality reduction [59]. This research achieved the highest accuracy of 97.8%.

Stance detection analysis has been performed for the purpose of fake news detection using web articles, studying their titles and their content in [60]. Authors have experimented with 4 different models, namely, ANN, LSTM, Bidirectional-LSTM, and CNN. Each model was tested with inputs of different kinds of embeddings (Tokenizer, GloVE, GloVE and attention mechanism, BERT). The combination of BERT embeddings with the CNN architecture produced the best results, with accuracy 95.32% and F1 score of 95.31%.

Also, a sequential approach developing the model Branch-LSTM was the model with the best performance for the SemEval-2017 Task 8 (Subtask A)[5]. This task examined a dataset of Twitter

[2] https://saifmohammad.com/WebPages/StanceDataset.htm

[3] https://alt.qcri.org/semeval2017/task8/index.php?id=data-and-tools

[4] http://www.fakenewschallenge.org/

[5] https://competitions.codalab.org/competitions/16171

conversations associated with rumours related to 10 different events, and achieved an accuracy of 0.784 [61].

3.2 GNN for data analysis over Social Media

Data from Twitter and Yelp have been used as a testbed for the application of baseline and GNN models, in a comparative study, [62]. Specifically, for the analysis of 4 Twitter datasets (fake news, hate speech, irony stereotype, stereotype stance) and a Yelp dataset of reviews, the researchers used three Transformer-based models (DistilBERT, ROBERTa, DistilROBERTa) and three GNN models (GraphSAGE, GAT, GraphTransformer). The results showed that in general GNN models had inferior performance compared to the baseline models except for one experiment, where a Graph Transformer model had better performance in the case of the hate speech dataset which was the smallest of all datasets. This result led to the conclusion that large datasets are better trained using a baseline model, while GNN perform better with smaller datasets.

A Graph Network model (GraphRec) has been proposed for social recommendation. The architecture takes in account interactions and opinions of users captured jointly in a user-item graph leading to the result that users' opinions play the most important role [63].

A combinational model of GNNs was proposed by Li et al., [64], for anomalous user detection. Different kind of anomalous users such as zombies (fake followers), bots and spammers are flooding social networks and their detection is a critical task. The three-layer model consisting of a GCN fusion layer, a GAT embedding layer, and a GNN aggregator of the two previous layers, presents the best performance among other individual models that analyzed data from Twitter and Yelp. The highest score was obtained in the analysis of the Twitter dataset in which the training data were the 30% of the dataset; achieved accuracy was 79.82% and recall was 71.84% [64].

A Multiple-aspect Attentional Graph Neural Network has been implemented for the purpose of geo-localization of users in social networks [65]. This model unifies the text of the posts and the interactions in the network to predict user location. It has been based on three Twitter dataset that contain users with location information. The model consists of an attention-based content learning, a GNN based on the interactions of users and a geological predictor and is able to predict user geolocation with higher accuracy than previous methods. This method has the advantage that it effectively combines features from both content and network with the use of a GNN.

From the perspective of fake news identification in social networks, a GNN model has been developed to investigate the pattern of propagation of fake news and real news [66]. This model is not based on the textual content but only on the features of the users, such as the number of followers, the number of friends, and the date of creation of a user's profile. The graph that has been created for this case represents the flow of the information. The root tweets are the tweets that shared a web article; following that, the tweets that retweeted the root message or other tweets create a cascade.

3.3 Previous studies on the 'SemEval-2017 Task 8 Subtask A' Dataset

The provided data set has been investigated by 8 different teams who submitted their approach for evaluation.

The proposed model by Kochkina [61] was the branch-LSTM that is depicted in figure 10, decomposing the tree structure of a thread into linear sequences.

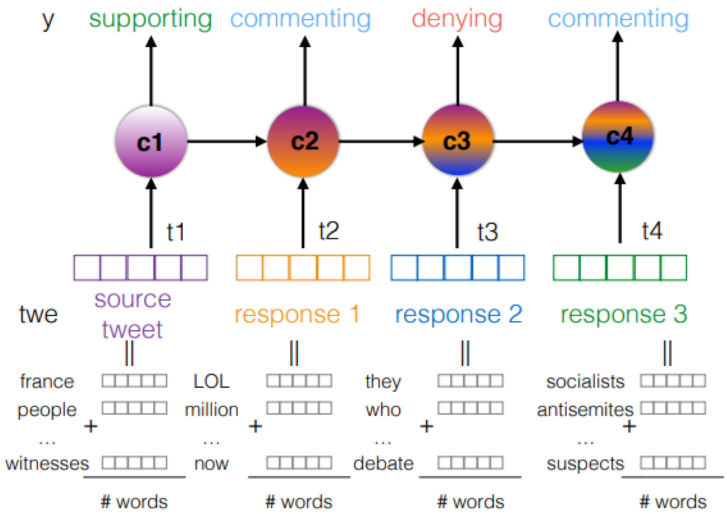

Figure 10: The proposed branch-LSTM architecture [61]

As illustrated in figure 10, at each time (t1, t2, t3, …), the input of LSTM model is a vector that represents a tweet providing a label to each tweet in a branch. The model was trained with different parameters such as different number of ReLU layers and different mini batch sizes. Finally, the set of parameters yielding the best performance achieved accuracy 0.784 while F1 score was 0.481. These results set the state-of-the art for rumour stance classification. A closer look into the results of the study, especially into the classification confusion matrix, revealed that the majority of the class 'Comment' was well predicted but other classes had more misclassifications than correct predictions. In specific, the model failed to make a single correct prediction for the class 'Deny'. It should be noted that the 'Comment' class is the bigger while the 'Deny' class is the smaller in this data set.

The second-best performance for the dataset was submitted by authors of [67]. The analysis was based on the training of Gradient Boosting model, a supervised classification algorithm. The accuracy of this model reached 78% with 0.45 F1 score. This study also faced problems with the prediction of 'Deny' class, predicting only 0.52% of the samples of this class.

The third-best model, [68], examined the dataset from the perspective of supervised machine learning methods and adopted a two-classifier solution in order to address the issue of data unbalance. The first classifier was built for discriminating whether a tweet is a comment or not (the majority of the tweets in this data set fall in the comment class), and the second classifier classified the tweet as belonging to any one of the rest of classes, in case it was not a comment. The algorithms that consistently performed well were the Logistic Regression (LR) and the Support Vector Machine (SVM). The best performance achieved an accuracy of 77.8%.

Another submission of analysis on the dataset investigated three different machine learning algorithms, [69]. Authors submitted the results of Maximum Entropy classification (MaxEnt) but they also studied other algorithms, such as the Naïve Bayes classifier and the Window classifier. The accuracy of the MaxEnt model reached 63.5%, but a combinational model of the three examined algorithms achieved 70.5%. Unfortunately, F1 scores are not reported in the study.

Other studies achieved accuracy of 64% using SVM classification, [70], 70.1% using Convolutional Neural Networks, [71], 74.9% using CNN architecture, [72], and 70.9% using Logistic Regression, [73]. It should be noted that all the above results refer to the testing data only.

3.4 SemEval-2019 Task 7 dataset

The above SemEval-2017 dataset have been enriched with additional data to produce the SemEval 2019 (Task A is related to stance detection), [74]. The version of 2017 includes totally 297 threads of 8 breaking news events. The set of data has been split into training, validation and testing data subsets. The whole 2017 data set was used as training data for the 2019 version which was augmented with more Twitter and Reddit data. This task had 22 submissions with the majority of them adopting neural networks approaches. The second-best performance achieved 85.50% accuracy and 61.67 F1 score using the BERT large model [75] while the best performance has been achieved through the methodologies of data expansion, data slicing, feature extraction and input concatenation and reached an F1 score of 61.87% on the test data [76]. The approach was based on fine-tuning of Generative Pretrained Transformer (OpenAI GPT). Later, another study outperformed the best submitted implementation and achieved an F1 score of 63.01% using a RoBERTa large model, [77].

4 CHAPTER 4: IMPLEMENTATION

4.1 Dataset

The dataset that has been used in this study is an open-access dataset[6] created for the investigation of rumours. Specifically, the data set consists of tweets pertaining to a particular event and rumours about it. It is published as 'SemEval-2017 Task 8: Rumour Eval: Determining rumour veracity and support for rumours' [78]. The dataset proposes two tasks: (i) stance detection and (ii) veracity of rumours. The current thesis focuses on the stance detection task.

The data collected are tweets that are referred to 8 well-known breaking news with the majority of them referring to the events of

- the Charlie-Hebdo shooting in Paris,
- shootings at Ottawa Parliament Hill,
- the Germanwings plane crash in the French Alps,
- a siege at a café in Sidney, and
- the unrest in Ferguson after a fatal shooting of an 18-year-old man by a police officer.

The tweets have been mined according to the facts' hashtags and keywords. The dataset is divided in two parts, the training set and the testing set, while the training data set includes a small subset of validation data.

1. The training data set contains 297 threads with 297 main tweets and their 4,222 replies, 4,519 tweets in total. Out of the 297 threads of the train data set, 25 are used for validation data.
2. The testing data set contains 28 additional threads. 20 threads are related to the previous rumours and 8 threads to new collected data related to other rumours, Hilary Clinton's pneumonia diagnosis and Youtuber Marina Joyce's kidnapping. The source 28 tweets include 1,052 replies, so the test data set consists of 1,080 tweets in total.

Each thread consists of a main tweet and its replies in a tree structure. Each tree consists of the source tweet as root and response tweets below that as branches. The tree can be expanded to more than one level as responses to other responses are included as well. Each tweet is annotated as one of 4 classes, namely, as:

1. *Support*: The content of the tweet supports the veracity of the rumour.
2. *Deny*: The content of the tweet denies the veracity of the rumour.
3. *Query*: The tweet contains a question about additional evidence related to the veracity of the rumour.

[6] The dataset is available on the link https://alt.qcri.org/semeval2017/task8/index.php?id=data-and-tools

4. *Comment*: The content of the tweet does not have any clear assessment related to the rumour and it is considered as a comment.

The data consists of folders of source tweets named by their IDs. Each source tweet folder contains the structure json file of the thread that depicts all the replies of the source tweet in a format of a dictionary (fig. 11). This structure illustrates which message is a reply to which message as the replies are contained in a sub-dictionary of the message they are addressed to.


```
{
    "498430783699554305": {
        "498432131669192704": {
            "498455038701092864": [],
            "498455687836758016": [],
            "498456361286795265": {
                "498465487635099648": [],
                "498471393538093056": []
            }
        },
        "498433698149056513": {
            "498488174923227136": []
        },
        "498440508256292864": {
            "498448178270990336": [],
            "498448625643831296": [],
            "498488821197991937": [],
            "498504798971650049": {
                "498521904396115971": []
            }
        },
        "498451864112074752": [],
        "498452460881870848": [],
        "498452989930401792": [],
        "498453364045529088": [],
        "498453873124970496": [],
        "498454395248721920": [],
        "498474828278804481": []
    }
}
```


Figure 11: Structure of a thread

Each source tweet folder also includes folders with the reply IDs and the content of them, again in a json format, as depicted in figure 12. There is a json file for each tweet of the data set that contains all extracted information.


```
"contributors": null,
"truncated": false,
"text": "@RT_com Micheal Brown was a hardened criminal. But , of course, shooting him was wrong and the cop should get the death penalty",
"in_reply_to_status_id": 500278045597368320,
"id": 500278685698519041,
"favorite_count": 0,
"source": "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>",
"retweeted": false,
"coordinates": null,
"entities": {
    "symbols": [],
    "user_mentions": [
        {
            "id": 64643056,
            "indices": [
                0,
                7
            ],
            "id_str": "64643056",
            "screen_name": "RT_com",
            "name": "RT"
        }
    ],
    "hashtags": [],
    "urls": []
},
"in_reply_to_screen_name": "RT_com",
"id_str": "500278685698519041",
"retweet_count": 0,
"in_reply_to_user_id": 64643056,
"favorited": false,
```


Figure 12: Content of a tweet

The data are manually annotated as 'support', 'deny', 'query' and 'comment' by journalists in the role of experts, and the dataset contains an individual json file in the format of a dictionary where keys represent the IDs and values represent the annotations (fig. 13).


```
"580321716838100992": "support",
"580332626541694976": "support",
"581065934598582272": "comment",
"580342887910494208": "query",
"581576082362478595": "comment",
"581174677554380800": "support",
"581473546200707073": "support",
"580341529870348288": "query",
"580339386262380544": "support",
"580361957502517248": "support",
"580321156508577792": "support",
```


Figure 13: Example of annotated data

4.2 Processing of the dataset

As the dataset consists of several separate files containing both their content and structure, a pre-processing step is necessary to gather all information that later can be processed to feed the neural network models. During the pre-processing of the data in Jupyter environment, a function is created to read the content of all tweet folders and match each tweet ID with its text, thus creating a data frame of all tweets. By reading the file with the annotations, each tweet ID is assigned the corresponding annotation. Each row of the data frame (fig. 14) contains the ID, the message and the stance annotations that is coded according to the table 1.

Table 1: Categorisation of stances in 4 numerical classes

Stance	Value
Support	0
Comment	1
Query	2
Deny	3

	id	tweet	stance
0	498280920227794945	@MichaelSkolnik -- wow	1
1	498282494387843072	@MichaelSkolnik Unbelieveable ..... We R in do...	1
2	498283789744103424	@MichaelSkolnik Emmett Till was visiting his f...	1
3	498323335965868032	"@MichaelSkolnik: Mike Brown was staying with ...	0
4	498432131669192704	@MichaelSkolnik cool! Darkskinned man is a cri...	1

Figure 14: A part of the created data frame of the dataset

The text of the messages contains different symbols, emojis and URLs that must be removed or replaced as they don't provide useful information. The symbol '@' means the mentioning of a user and in the pre-processing step will be replaced by the term 'user'. For example, the term '@MichaelSkolnik' will be replaced by 'user'. Respectively, all the URLs in the messages will be replaced with the term 'url' and popular emojis are replaced by the emotion they express, such as 'smile', 'heart', and 'sadface'. The text pre-processing transforms all character into lowercase to avoid any confusion between uppercase and lowercase characters. Then, the messages are tokenized so the text is replaced by a list which contains the tokens of the words. The next step includes the creation of the embeddings using Word2Vec model. The size of the Word2Vec has been set to 200. After passing the tokens to the model, a vector of embeddings is created for each sentence. As the length of the vectors varies, an average of the vectors is calculated, so finally each sentence is represented by a vector of 200 values.

During the pre-processing of the data, all structure files have been converted to dictionaries, so each thread is represented by two lists:

- a list of sources, and
- a list of targets,

where source is a message and target is the reply to it (fig. 15). This dictionary helps to contain the edges of each thread for the creation of the graphs later.

```
{'source_list': ['500280422295937024', '500280422295937024', '500280422295937024', '500280422295937024', '50028042229
5937024', '500280422295937024', '500280422295937024', '500280422295937024', '500280422295937024', '50028042229593702
4', '500280422295937024', '500289950735400961', '500290872479449089'], 'target_list': ['500281136439103489', '5002819
06320404480', '500282011982913539', '500282657037496320', '500283535769759744', '500283809989132288', '50028394408942
3872', '500284024162488321', '500284169382289409', '500289950735400961', '500339373867827201', '500290872479449089',
'500291568361041920']}
```

Figure 15: Dictionary of edges of a single thread

The tweet IDs are transformed into numerical representation from 0 (source tweet) and consecutive numbers for the rest of the tweets. Finally, a dictionary is created that contains the nodes, the edges, the stance value and the embeddings for each thread (fig. 16).

```
{'tweet_id': [498280126254428160,
  498280920227794945,
  498282494387843072,
  498283789744103424,
  498323335965868032],
 'node_id': [0, 1, 2, 3, 4],
 'stance': [0, 1, 1, 1, 0],
 'edges': [[0, 0, 0, 0], [1, 2, 3, 4]],
 'embeddings': [array([-0.5289016 ,  0.6455914 ,  0.05664925, -0.33407387,  0.18608259,
        -0.6590995 ,  0.43578324,  0.99567777,  0.00694965, -0.4032204 ,
        -0.12305938, -1.204547  , -0.45057458,  0.31459337,  0.12246821,
        -0.39512047, -0.28556457, -0.7299466 ,  0.27002144, -0.8171757 ,
         0.03485603,  0.54994935,  0.5354931 ,  1.0957217 , -0.69440323,
         0.01716423, -0.6067445 , -0.76181513,  0.03859043, -0.01284096,
         0.45732394, -0.05438009,  0.3311227 , -0.30755463,  0.32867122,
         0.7600045 ,  0.10541353, -0.636778  , -0.77018076, -0.7001457 ,
         0.30077145, -0.28526264,  0.15013562, -0.05170989,  0.5697572 ,
```

Figure 16: Data dictionary for each thread

This format can feed a function that uses Python NetworkX package to create graphs such as those of figure 17. During the creation of the graphs, replies above the first level of replies, are also connected to the source message. In the following graphs, node 0 represents the source tweet and the rest of the nodes represent the replies.

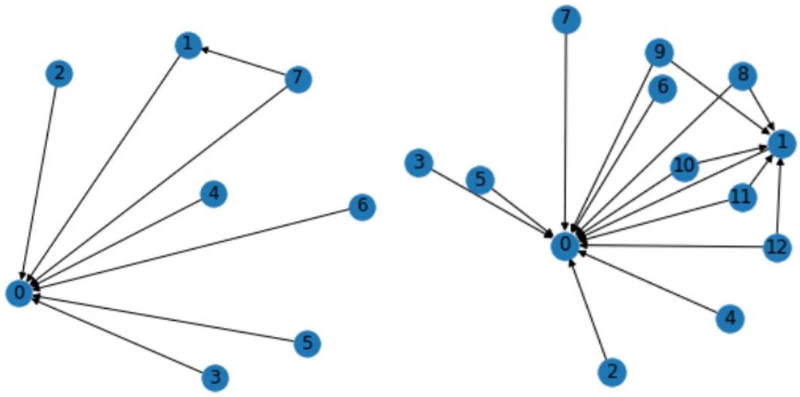

Figure 17: Graphs of threads

4.3 Methodology

The methodology that has been employed in this thesis concerns two models: a pure GCN model and a combinational model of GCN and LSTM.

The library that has been used for the processing of the data is the Deep Graph Library (DGL). Itis a Python package built for implementation of GNNs. The model that has been used is the GCN model with initially two convolutional layers. The ReLU has been used as activation function. The first architecture of the GCN model uses Word2Vec to transform the words into vectors of 200 dimensions; the window size is set to 3. Window size is the number of words that are taken in account before and after a given word as its 'context'. The parameter 'min_count' is set to 5 which is the default value of the model. This parameter refers to the minimum number of times that a word has to appear in the corpus so as not to be dropped from the vocabulary during training. The model initially implemented has two convolution layers; results are obtained with additional layers, too. The pipeline of this model in a schematic diagram is depicted in figure 18.

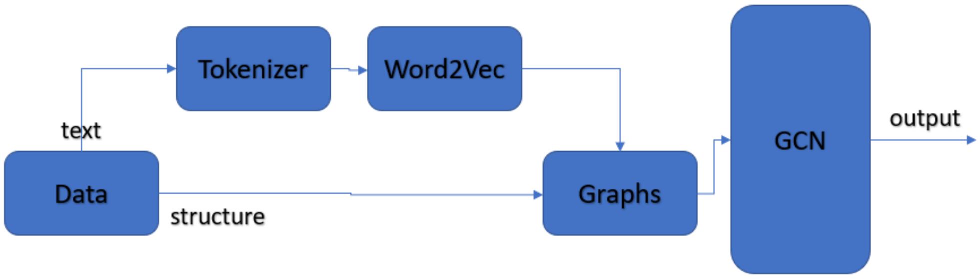

Figure 18: Pipeline of GCN model

The second model is the combination of LSTM and GCN. This time the text is represented not by Word2Vec embeddings but by the features produced during training of the LSTM. The architecture is depicted in figure 19.

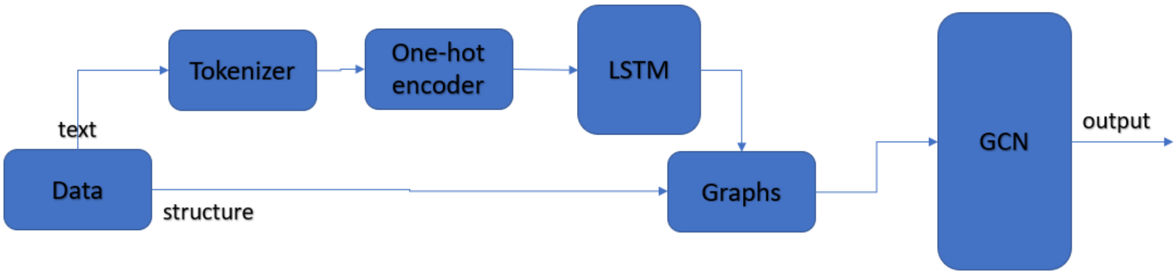

Figure 19: Pipeline of GCN-LSTM model

Initially, the text is tokenized with the number of tokens set to 40 for all texts and using a padding sequence for the shorter texts. The embeddings with the format of one-hot encoding feed the LSTM model. The model computes the features during training. Efficiency is measured during each epoch but only the model of the epoch with the best performance is saved. The features of this model are embedded as features in the graphs which in turn feed the GCN.

5 CHAPTER 5: RESULTS

The first model that was implemented with GCN reached an accuracy value of 77.48% and an F1 score of 0.4291. The model performed better with 5 convolutional layers while the activation function that was used was ReLU.

This first architecture was tested with different parameters, such as different number of convolution layers and different parameters of the Word2Vec model. During training, the model saved the version of the epoch with the best performance. Table 2 depicts the different versions of the model, according to the different number of layers, for a Word2Vec vector size of 200.

Table 2: Accuracy of the model with different number of Convolutional layers

GCN Layers	Accuracy	F1 score
2	77.18%	0.41172
3	76.88%	0.4143
4	76.09%	0.4126
5	77.48%	0.4291

The confusion matrix that reveals in details the predictions for each one of the four classes of the task is depicted in figure 20.

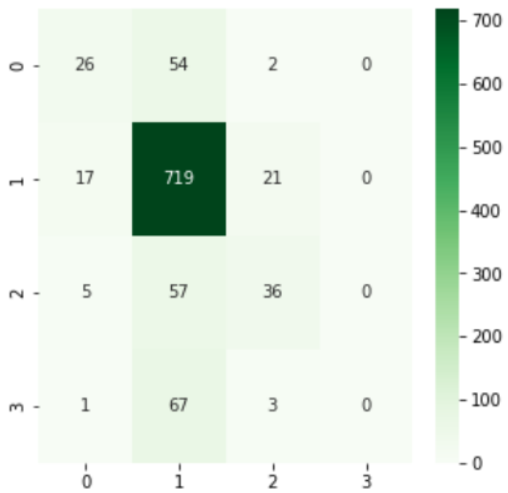

Figure 20: Confusion Matrix of the GCN that had the best performance

This confusion matrix shows that class 1 (‘comments’) is the most efficiently predicted class, while for other classes prediction has significant loss in performance. It is noteworthy that class 3 (‘deny’) is not predicted at all. Classes 0 (‘support’) and 2 (‘query’) get far less than 50% of correct predictions.

The above model of 5 layers has been tested with 2 other sizes of embedding vectors, higher than 200:

- with a dimension of Word2Vec vectors equal to 500, accuracy reached 75.89%, and F1 score = 0.4014.
- with a dimension of Word2Vec vectors equal to 800, accuracy reached 75.60% and F1 score = 0.3839.

The GCN-LSTM model did not produce better results. The accuracy of this model reached 61.31% with F1 score = 0.2447, for two convolution layers. The heatmap of this model is depicted in figure 21.

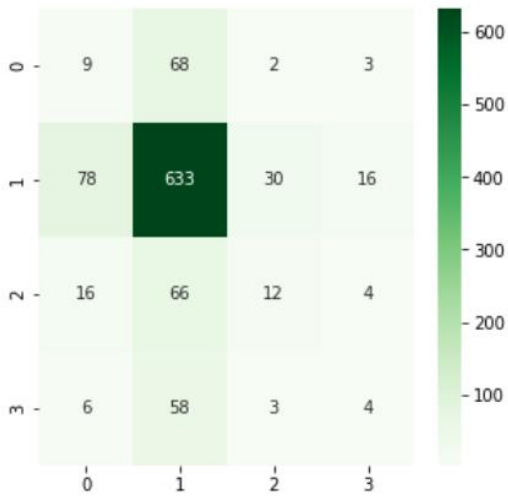

Figure 21: Confusion Matrix of the LSTM-GCN model

Table 3 depicts the results of previously submitted studies who were based on the same data set, namely, RumourEval-2017. Evaluation results of previously submitted models are found in publication [78], in the form of accuracy scores. Models are shown in decreasing order of accuracy. The results of the current thesis (GCN model) have been added in the same table for comparison; they hold the 4th place, by a margin of 0.01 from the results holding the 1st place.

Table 3: Ranking table of previously submitted results and current thesis results, obtained on the same data set.

Team	Accuracy
Turing	0.784
UWaterloo	0.780
ECNU	0.778
Current Thesis	0.774
Mama Edha	0.749
Nile TMRG	0.709
IKM	0.701
IITP	0.641
DFKT DKT	0.635

6 CHAPTER 6: CONCLUSIONS – FUTURE WORK

6.1 Conclusions

The rapid development of social media and their daily use has raised the interest of researchers to study the behavior of users; research results lead to conclusions from various aspects. A critical aspect is misinformation: indeed, it can lead to harmful situations for the society, especially in periods of crises, such as the recent COVID-19 pandemic during which unscientific articles and messages were widely disseminated through social networks (among other media). The urgent need to develop tools that can investigate people's behavior, opinion, stance and attitudes led to the development of techniques that provide useful insights exploiting the big data of online social networks.

Natural Language Processing includes various methods for analyzing data to extract useful information, such as the sentiment of people as expressed in social media, the attitude of people towards other users' posts and how objectively or subjectively people react on messages. The interest of researchers led to a plethora of architectures using Machine Learning and Deep Learning techniques, proposed to examine users' behavior, attitude or stance, such as Support Vector Machine, Naïve Bayes Classifier, and neural networks architectures such as CNN and RNN.

Existing research studies have examined data mined from social network platforms, such as Twitter and Reddit, focusing mainly on specific issues of misinformation. Different data sets have been analyzed for this purpose. This thesis examines the SemEval-2017 data set focused on rumors on topics of hot public interest. The data set was analyzed from the perspective of stance detection; the goal was to develop a predictive model based on the GNN architecture. For this reason, the GNN architecture was first described, along with important techniques used for this purpose in the context of text processing.

Two models have been implemented for the analysis of this thesis, a pure GCN model and a combinational model of an LSTM and GCN architecture. The highest prediction accuracy of these implementations was obtained by the pure GCN model with 5 convolutional layers; this setup reached an accuracy of 77.48% with F1 score = 0.429. Despite the fact that this result is higher than similar results obtained in existing studies, it does not outperform the best existing accuracy which is 78.4% with F1 score = 0.481.

The approach of the pure GCN model seemed to be fairly close to this highest accuracy achieved today. On the other hand, previous studies have expressed concern about the unbalanced nature of the specific data set, especially as it became clear that the overall high accuracy was achieved mainly thanks to the high score in the class of 'comments', which was the bigger class, while all smaller classes got considerably lower scores.

The implementation of the combination LSTM-GCN model didn't produce satisfactory results. The disproportionate numerical superiority of one class over all other classes, as was the case with the specific data set, is certainly a negative factor in getting good prediction scores for underpopulated classes. On the other hand, this fact should not inhibit further research for a model that can successfully face this issue and offer satisfactory results on unbalanced data sets.

6.2 Future Work

As explained in the previous paragraph, the unbalanced data set is considered as the major cause of the low efficiency exhibited by the 2 investigated models, especially in the underpopulated classes. This observation gives rise to a first thought for future research work towards expanding the data set. The enrichment of the data set was already conducted by its authors, who created the next version, namely, RumourEval-2019. This version includes additional data from Twitter and Reddit. Future research could proceed to the investigation of this expanded data set using the architectures proposed of this thesis and aiming to derive enhanced results. It would be interesting to comparatively evaluate the efficiency of GNNs in the stance detection task and check if the new, enriched and better-balanced data set would lead to a higher performance, in comparison with results obtained in previous studies.

Another suggestion is the adoption of a different GNN architecture, such as GAT, in order to compare the effectiveness of these two approaches. It would also be interesting to approach the task using a Large Language Model (LLM). Especially the combination of LLMs and GNNs is a topic of keen research interest, [79], as the issue of limited training data can be counterbalanced by the extensive knowledge of the LLMs. Specifically, integrating LLMs into graph algorithms can provide enhanced mechanisms of the features attached to nodes of the graph, thus improving the performance of the GNN model.

As a final comment, the improvement of GNN architectures is of high research interest because they can be adopted and introduced in various and different methodologies and offer sometimes innovative findings, such as the discovery of techniques for new antibiotics in medicine, [80]. From the perspective of misinformation, a source of concern is the evolution of AI: indeed, AI could create persuasive fake news that may put people at risk. Researchers must therefore be alert and ready to exploit state-of-the-art solutions to address this phenomenon, for the prosperity of the society.

Bibliography – References – Online sources

[1] "Eurobarometer: Media & News Survey 2022". https://europa.eu/eurobarometer/surveys/detail/2832 (accessed Jan. 25, 2024).

[2] "Special Eurobarometer 516 : European citizens' knowledge and attitudes towards science and technology" (v1.00). (2021). [Data set]. European Commission, Directorate-General for Communication. http://data.europa.eu/88u/dataset/S2237_95_2_516_ENG.

[3] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election", Journal of Economic Perspectives—Volume 31, Number 2—Spring 2017—Pages 211–236.

[4] D. Mahl, M.S. Schäfer and J. Zeng, "Conspiracy theories in online environments: An interdisciplinary literature review and agenda for future research", 2023, New Media & Society, 25(7), 1781–1801. https://doi.org/10.1177/14614448221075759.

[5] S. Naeem, R. Bhatti, & A. Khan, "An exploration of how fake news is taking over social media and putting public health at risk", Health Information &Amp; Libraries Journal, vol. 38, no. 2, p. 143-149, 2020. https://doi.org/10.1111/hir.12320.

[6] K. Shu, A. Sliva, S. Wang, J. Tang, & H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective", ACM SIGKDD Explorations Newsletter, Volume 19, Issue 1, pp 22-36, https://doi.org/10.1145/3137597.3137600.

[7] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, & R. Procter, "Detection and Resolution of Rumours in Social Media: A Survey", 2018. ACM Computing Surveys 51, 2, Article 32 (February 2018), https://doi.org/10.1145/3161603.

[8] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, & J. Han, "A Survey on Truth Discovery", 2015. arXiv:1505.02463v2. https://doi.org/10.48550/arXiv.1505.02463.

[9] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, ""8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality", AAAI, vol. 30, no. 1, Feb. 2016. DOI: https://doi.org/10.1609/aaai.v30i1.9966.

[10] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots", Communications of the ACM, Volume 59, Issue 7, pp 96–104, https://doi.org/10.1145/2818717.

[11] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification", 2018, Security and Privacy, Vol. 1, Issue 1, https://doi.org/10.1002/spy2.9.

[12] X. Zhou, and R. Zafarani, "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities", ACM Computing Surveys, Volume 53, Issue 5, Article No.: 109, pp 1–40, https://doi.org/10.1145/3395046.

[13] M. Aldwairi and A. Alwahedi, “Detecting Fake News in Social Media Networks”, Procedia Computer Science, Volume 141, 2018, Pages 215-222, https://doi.org/10.1016/j.procs.2018.10.171.

[14] Alessandro Bondielli, and Francesco Marcelloni, “A survey on fake news and rumour detection techniques”, Information Sciences, Volume 497, 2019, Pages 38-55, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2019.05.035.

[15] C. N. Kamath, S. S. Bukhari, A. Dengel, “Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification”, Proceedings of the ACM Symposium on Document Engineering 2018, August 2018, Article No.: 14, Pages 1–11, https://doi.org/10.1145/3209280.3209526.

[16] M.A. Alonso, D. Vilares, C. Gómez-Rodríguez, and J. Vilares, “Sentiment Analysis for Fake News Detection”. Electronics. 2021; 10(11):1348. https://doi.org/10.3390/electronics10111348.

[17] K.R., Chowdhary. “Natural Language Processing. In: Fundamentals of Artificial Intelligence”. 2020, Springer, New Delhi. https://doi.org/10.1007/978-81-322-3972-7_19.

[18] Shiliang Sun, Chen Luo, Junyu Chen, “A review of natural language processing techniques for opinion mining systems”, 2017, Information Fusion, Volume 36, Pages 10-25, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2016.10.004.

[19] W. Khan, A. Daud, J. Nasir, and T. Amjad. “A survey on the state-of-the-art machine learning models in the context of NLP”, 2016, Vol.43, No 4 (2016): Kuwait Journal of Science.

[20] O. Melamud, D. McClosky, S. Patwardhan, and M. Bansal, “The Role of Context Types and Dimensionality in Learning Word Embeddings”, 2016, In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1030–1040, San Diego, California. Association for Computational Linguistics. 10.18653/v1/N16-1118.

[21] S. Selva Birunda, and R. Kanniga Devi, R, “A Review on Word Embedding Techniques for Text Classification”. 2021, In: Raj, J.S., Iliyasu, A.M., Bestak, R., Baig, Z.A. (eds) Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies, vol 59. Springer, Singapore. https://doi.org/10.1007/978-981-15-9651-3_23.

[22] S. Khomsah, R. D. Ramadhani, and S. Wijaya, “The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews”. 2022. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 6(3), 352 - 358. https://doi.org/10.29207/resti.v6i3.3711.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space”, 2013. arXiv:1301.3781v3, https://doi.org/10.48550/arXiv.1301.3781.

[24] D. Jatnika, M. A. Bijaksana, and A. Ardiyanti Suryani, "Word2Vec Model Analysis for Semantic Similarities in English Words", Procedia Computer Science, Volume 157, 2019, Pages 160-167, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.08.153.

[25] https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e. (accessed Feb. 19, 2024).

[26] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into Deep Learning". https://d2l.ai/index.html (accessed Feb. 19, 2024).

[27] Van Houdt, G., Mosquera, C. & Nápoles, G. A review on the long short-term memory model. Artif Intell Rev 53, 5929–5955 (2020). https://doi.org/10.1007/s10462-020-09838-1

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, & I. Polosukhin, 2017, "Attention is all you need", 2017, arXiv:1706.03762v7, https://doi.org/10.48550/arXiv.1706.03762.

[29] D. Rothman, "Transformers for Natural Language Processing", 2021, Packt Publishing Ltd., ISBN 978-1-80056-579-1.

[30] J. Devlin, M.-W. Chang, K. Lee, & K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2019, arXiv:1810.04805v2,

https://doi.org/10.48550/arXiv.1810.04805

[31] M. Szczepański,, M. Pawlicki, R. Kozik, & M. Choras. "New explainability method for BERT-based model in fake news detection" 2021, Sci Rep 11, 23705. https://doi.org/10.1038/s41598-021-03100-6.

[32] A. Shrestha, M. L. Tlachac, R. Flores, & E. A. Rundensteiner. "BERT Variants for Depression Screening with Typed and Transcribed Responses" 2023. In Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers (UbiComp/ISWC '22 Adjunct). Association for Computing Machinery, New York, NY, USA, 211–215. https://doi.org/10.1145/3544793.3563405

[33] W. L. Hamilton, "Background and Traditional Approaches. In Graph Representation Learning", 2020 Springer International Publishing, Cham, 9-27. https://doi.org/10.1007/978-3-031-01588-5_2.

[34] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: a model and comprehensive taxonomy", 2022, The journal of Machine Learning Research, Vol. 23, Issue 1, ISSN: 1532-4435.

[35] P. Latouche and F. Rossi, "Graphs in Machine Learning: an introduction", 2015, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Proceedings, ISBN: 978-2-87587-014-8.

[36] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications”, AI Open, Volume 1, 2020, Pages 57-81, ISSN 2666-6510, https://doi.org/10.1016/j.aiopen.2021.01.001.

[37] F. Scarselli, A.C. Tsoi, M. Gori, and M. Hagenbuchner. “Graphical-based learning environments for pattern recognition”. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR /SPR 2004. LNCS, vol. 3138, pp. 42–56. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27868-9_4

[38] W. Hamilton, “Graph Representation Learning”, 2020, Synthesis Lecture on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, ISBN: 9781681739649, DOI: 10.2200/S01045ED1V01Y202009AIM046

[39] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, and G.E. Dahl, “Neural message passing for quantum chemistry”. 2017, Proceedings of the 34th International Conference on Machine Learning, PMLR 70:1263-1272, arXiv:1704.01212.

[40] Z. Huang, W. Chung, T. Ong, and H. Chen, “A graph-based recommender system for digital library”, JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, July 2002, Pages 65–73, https://doi.org/10.1145/544220.544231.

[41] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks”, 2018, Bioinformatics, 34:13, 457-466, 2018, arXiv:1802.00543, https://doi.org/10.1093/bioinformatics/bty294

[42] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. “Translating embeddings for modeling multi-relational data”. 2013. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[43] S. Pandit, D. Chau, S. Wang, and C. Faloutsos. “NetProbe: A fast and scalable system for fraud detection in online auction networks”. WWW '07: Proceedings of the 16th international conference on World Wide Web, May 2007, Pages 201–210, https://doi.org/10.1145/1242572.1242600.

[44] M. Agrawal, M. Zitnik, and J. Leskovec, “Large-scale analysis of disease pathways in the human interactome”. 2018. Biocomputing 2018, pp. 111-122 (2018).

[45] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli. “Graph matching networks for learning the similarity of graph structured objects”. In ICML, 2019, 3835-3845.

[46] T.N. Kipf, and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks”, 2017, ICLR 2017, arXiv:1609.02907v4, https://doi.org/10.48550/arXiv.1609.02907.

[47] Graph Convolutional Network. https://ai.plainenglish.io/graph-convolutional-networks-gcn-baf337d5cb6b (accessed 27 Feb, 2024).

[48] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, "Graph Attention Network", (2018), ICLR 2018, arXiv:1710.10903v3, https://doi.org/10.48550/arXiv.1710.10903.

[49] W. Hamilton, R. Ying, J. Leskovec, "Inductive Representation Learning on Large Graphs", 2017, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Pages 1025–1035

[50] D. Biber and E. Finegan, "Adverbial stance types in English", Discourse processes, vol. 11, no 1, pp. 1–34, 1988 https://doi.org/10.1080/01638538809544689

[51] A. AlDayel and W. Magdy, "Stance Detection on Social Media: State of the Art and Trends", Information Processing & Management, Volume 58, Issue 4, 2021, 102597, ISSN 0306-4573 2020, arXiv:2006.03644v5, https://doi.org/10.48550/arXiv.2006.03644.

[52] Y. Hou, P. V. de Putten and S. Verberne, "The COVMis-Stance dataset: Stance Detection on Twitter for COVID-19 Misinformation", 2022, arXiv:2204.02000v1, https://doi.org/10.48550/arXiv.2204.02000

[53] M. Lai, A. T. Cignarella, D. I. H. Farías, C. Bosco, V. Patti and P. Rosso, "Multilingual stance detection in social media political debates", Computer Speech & Language, Volume 63, 2020, 101075, ISSN 0885-2308, https://doi.org/10.1016/j.csl.2020.101075.

[54] S. Somasundaran, J. Wiebe, "Recognizing stances in online debates", Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Association for Computational Linguistics, Stroudsburg, PA, USA (2009), pp. 226-234.

[55] D. Küçük and F. Can, "Stance Detection on Tweets: An SVM-based Approach", 2018, arXiv:1803.08910v1, https://doi.org/10.48550/arXiv.1803.08910.

[56] W. Wei, X. Zhang, X. Liu, W. Chen and Tengjiao Wang, "pkudblab at SemEval-2016 Task 6: A Specific Convolutional Neural Network System for Effective Stance Detection", Proceedings of the 10th International Workshop on Semantic Evaluation ({S}em{E}val-2016), 2016, Association for Computational Linguistics, pp. 384-388, doi: 10.18653/v1/S16-1062.

[57] A. Zubiaga, E. Kochkina, M. Liakata, R. Procter, M. Lukasik, K. Bontcheva, T. Cohn, and I. Augenstein, "Discourse-aware rumour stance classification in social media using sequential classifiers", Information Processing & Management, Volume 54, Issue 2, 2018, Pages 273-290, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2017.11.009.

[58] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Marquez and A. Moschitti, "Automatic stance detection using End-to-End memory networks", arXiv:1804.07581, 2018, [online] Available: `http://arxiv.org/abs/1804.07581`.

[59] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi and B. -W. On, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," in IEEE Access, vol. 8, pp. 156695-156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

[60] H. Karande, R. Walambe, V. Benjamin, K. Kotecha, T. S. Raghu, "Stance Detection with BERT Embeddings for Credibility Analysis of Information on Social Media", 2021, arXiv:2105.10272v1, https://doi.org/10.48550/arXiv.2105.10272.

[61] E. Kochkina, M. Liakata, I. Augenstein, "Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM". In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC,Canada, 3–4 August 2017; pp. 475–480. doi:10.18653/v1/S17-2083.23.

[62] M. Toshevska, S. Kalajdziski and S. Gievska, "Graph Neural Networks for Antisocial Behavior Detection on Twitter", 15th ICT Innovations International Conference 2023, arXiv:2312.16755, https://doi.org/10.48550/arXiv.2312.16755

[63] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation", WWW '19: Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, May 2019, https://doi.org/10.1145/3308558.3313488

[64] Y. Li, Y. Ji, S. Li, S. He, Y. Cao, Y. Liu, H. Liu, X. Li, J. Shi, and Y. Yang, "Relevance-Aware Anomalous Users Detection in Social Network via Graph Neural Network," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534136.

[65] T. Zhong, T. Wang, J. Wang, J. Wu and F. Zhou, "Multiple-Aspect Attentional Graph Neural Networks for Online Social Network User Localization," in *IEEE Access*, vol. 8, pp. 95223-95234, 2020, doi: 10.1109/ACCESS.2020.2993876.

[66] Y. Han, S. Karunasekera and C. Leckie, "Graph Neural Networks with Continual Learning for Fake News Detection from Social Media", 2020, arXiv:2007.03316

[67] H. Bahuleyan, and O. Vechtomova, "UWaterloo at SemEval-2017 Task 8: Detecting. Stance towards Rumours with Topic Independent Features". 2017. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 461–464, Vancouver, Canada. Association for Computational Linguistics.

[68] Feixiang Wang, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 Task 8: Rumour Evaluation Using Effective Features and Supervised Ensemble Models. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 491–496, Vancouver, Canada. Association for Computational Linguistics.

[69] Ankit Srivastava, Georg Rehm, and Julian Moreno Schneider. 2017. DFKI-DKT at SemEval-2017 Task 8: Rumour Detection and Classification using Cascading Heuristics. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 486–490, Vancouver, Canada. Association for Computational Linguistics.

[70] Vikram Singh, Sunny Narayan, Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2017. IITP at SemEval-2017 Task 8: A Supervised Approach for Rumour Evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 497–501, Vancouver, Canada. Association for Computational Linguistics.

[71] Y-C. Chen, Z-Y. Liu, and H-Y. Kao, "IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification". 2017. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 465–469, Vancouver, Canada. Association for Computational Linguistics.

[72] M. G. Lozano, H. Lilja, E. Tjörnhammar, and M. Karasalo, "Mama Edha at SemEval-2017 Task 8: Stance Classification with CNN and Rules". 2017. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 481–485, Vancouver, Canada. Association for Computational Linguistics.

[73] O. Enayet and S. R. El-Beltagy. "NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter." 2017. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 470–474, Vancouver, Canada. Association for Computational Linguistics.

[74] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, and L. Derczynski. "SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours". 2019. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

[75] M. Fajcik, P. Smrz, and L. Burget, "BUT-FIT at SemEval-2019 Task 7: Determining the Rumour Stance with Pre-Trained Deep Bidirectional Transformers". 2019. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 1097–1104, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

[76] R. Yang, W. Xie, C. Liu, and D. Yu. "BLCU_NLP at SemEval-2019 Task 7: An Inference Chain-based GPT Model for Rumour Evaluation". 2019. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 1090–1096, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

[77] P. Kasnesis, L. Toumanidis, and C. Z. Patrikakis. "Combating Fake News with Transformers: A Comparative Analysis of Stance Detection and Subjectivity Analysis". 2021. Information 12, no. 10: 409. https://doi.org/10.3390/info12100409.

[78] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga, "SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours", 2017. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

[79] S. Pan, Y. Zheng and Y. Liu, "Integrating Graphs With Large Language Models: Methods and Prospects," 2024. IEEE Intelligent Systems, vol. 39, no. 1, pp. 64-68, Jan.-Feb. doi: 10.1109/MIS.2023.3332242.

[80] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay, and J. J. Collins. (2020). "A Deep Learning Approach to Antibiotic Discovery". Cell, 180(4), 688–702.e13. https://doi.org/10.1016/j.cell.2020.01.021

Appendix

The code developed for this thesis is available in the following link:

https://github.com/panagiotismon/stance_detection_rumoureval2017