INDIVIDUAL DIPLOMA THESIS

Stylianos Herodotou

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

May 2022

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

ENDURE: A Framework for Identifying Fake News Through Polarization Knowledge

Stylianos Herodotou

Supervisor:
George Pallis

The Individual Diploma Thesis was submitted for partial fulfilment of the requirements for obtaining the degree of Computer Science of the Department of Computer Science of the University of Cyprus

May 2022

Acknowledgements

I would like to express my gratitude to my supervisor, Associate Professor Dr George Pallis for his guidance and mentorship throughout this project, as well as throughout my overall studies at the University of Cyprus.

Moreover, I would like to thank Ph.D. Candidate, Demetris Paschalidis for his excellent guidance. I am grateful to him for his support and advice which played significant role in the fulfilment of this work

Finally, I would like to express my deepest appreciation to my family for their invaluable help during my studies. Specifically, I would like to thank my parents Herodotos and Eleni, and my sweetheart Despiana for the support they gave me. Words cannot express my gratitude to you.

Abstract

Fake news represents a significant threat to democracy and free speech. It weakens public trust in democratic institutions and distorts electoral processes. Polarization has been increasing dangerously over the last decade, exacerbating the effects of misinformation even further.

Despite the clear correlation between fake news and polarization, there has been no research so far that attempts to use polarization knowledge to aid in the detection of fake news. This work aspires to fill this void through ENDURE, a framework for generating, modelling and utilizing polarizing knowledge to identify Fake News in any domain.

We initially present our new Covid19 Fake News Dataset, containing over 161 thousand articles labelled as reliable or unreliable based on the reputation of their hosts. Next, we present how we model polarization, and we showcase four novel graph representation schemas, each highlighting a different aspect of polarization.

We continue by exploring the architecture of the ENDURE and we show its potential as a standalone Fake News Detection model. Our results show that it can achieve competitive performance, similar to LSTM based models.

Finally, we experiment with the idea of embedding our approach with existing state of the art models from the literature and found a noticeable increase in performance, with RoBERTa based model seeing an increase of +4% accuracy and LSTM based model increasing by a stunning +11% accuracy.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The term “Fake News”, first documented in the 1890s [98] and then popularized by Craig Silverman[97], refers to the deliberate spread of misinformation.

Needless to say, lies and deceit are not a recent phenomenon. The same is true for False and distorted news material. Nazi propaganda machines used Fake News to incite anti-Semitism and racist prejudice in the United States in the 1800s led to the printing of false stories about African Americans' supposed flaws and crimes. [100]

Moreover, in the 1890s, two newspaper publishers, Joseph Pulitzer and William Hearst, competed for the attention of the public by sensationalizing stories and publishing rumours as facts. Their incredulous news contributed in leading the US into the Spanish-American War in 1898 [99]

Because of the speed with which fake news is spread and the scale of its influence, today's fake news is fundamentally different from its pre-social media counterparts. Bovet et al [101] examined tweets in the five months preceding the 2016 US election

and found that 25% of the tweets of their dataset, spread either fake or extremely biased news.

Höller [102] manually fact-checked Britain's Political leaders using fact-checking platforms and found that they too took part in spreading the Fake News surrounding Brexit. In particular he concludes that "Boris Johnson and Nigel Farage shared multiple arguments that were clearly misleading".

Fake news is so prominent that Oxford Dictionaries declared the word "Post-Truth" to be 2016's word of the year. [102].

Fake news is an imminent threat to democracy and freedom of expression. It erodes trust in democratic institutions and distorts electoral processes. [1,2] Fake news not only misleads people into accepting false beliefs but also alters the way people respond to the truth. [4,5]. Moreover, it fosters incivility and polarisation.[1] The polarization increase in the last years has severely divided our societies. It has been observed on several high-profile occasions, from the US presidential elections, to the refugee crisis, to the Brexit referendum. It reached the point of violence at the storming of the United States Capitol, and the current pandemic of Covid19 displayed the alarmingly elevated levels of polarization over science even after overwhelming scientific evidence and consensus.[6]

1.2. Framework Overview

Although polarization and misinformation are undeniably linked, the direction of that link is less evident. The elevated levels of polarization either social or political provide fertile ground for the proliferation of fake news. On the other hand, misinformation could be said to exacerbate and magnify polarization. [1,10] Nevertheless, both polarization and misinformation are considered disturbing threads, and any attempt to mitigate one should coincide with the understanding of the other.

This work aims to quantify the relation between polarization and misinformation. As a compact and comprehensive approach, we model this task as a classification problem, identifying news articles as reliable or unreliable, while using polarization knowledge as input.

To this end, we propose ENDURE, a framework for generating, modelling and utilizing polarizing knowledge to identify Fake News in any domain.

Despite the clear correlation between fake news and polarization, there has been no research so far that attempts to use polarization knowledge to aid in the detection of fake news. This work aims to fill this void. To the best of our knowledge, this study describes the first attempt to detect fake news using Polarization information.

We identified the POLAR framework [24] to be the most effective at capturing the essence of Polarization in a collection of articles in an unsupervised manner. This work is an extension of the POLAR framework which makes use of the Polarization data it generates.

POLAR first constructs the Sentiment Attitude Graph, a signed undirected graph with key entities as nodes and their interactions as edges. It then identifies clusters of entities named Fellowships and polarized relationships between fellowships, called Fellowship Dipoles. Finally, it extracts the discussion topics of Fellowship Dipoles and quantifies each topic’s polarization.

ENDURE initially collects the content of public news articles that match the specified subject and time frame. It then Proceeds to use the POLAR framework [24] to generate the Polarization information of the fetched articles and produce a knowledge base of Public Polarization Information. In addition to that, the framework will also generate the polarization information of each article of the specified private dataset.

We have examined several different graph representations of the generated information, and we have identified four novel representation Schemas of the polarization data. Each Shema highlights a different aspect of the knowledge and allows the use of different techniques to express this knowledge in the desired downstream task. Moreover, have experimented with a variety of state-of-the-art approaches to generating embeddings for each one.

The two most expressive schemas are alternate versions of a directed heterogeneous graph that allows the use of different node and edge types. One allows for the use of node and edge features, while the other only node features. We have experimented with a selection of both deep and shallow encoders for these representations, with an emphasis on Graph Neural Networks.

Next, we have identified a schema in the format of a knowledge graph using triples of (subject, predicate, object) to captures the meaning of the polarization relationships. We have experimented with a variety of Knowledge graph embeddings techniques using both Translational distance and Semantic matching models.

Finally, we use a rawer representation of the polarization knowledge using only the Sentiment Attitude Graph. This raw representation of the data gives us the opportunity of including additional polarization information generated by other state of the art approaches based on balance theory. Moreover, it enables deep neural networks to find the optimal features for this problem [53].

ENDURE produces node embeddings for the chosen representation of the public knowledge base as mentioned above. After that we choose a graph representation for our private articles and embed the knowledge of the of Public Polarization Information into each one using the analogous node embeddings.

Next, we use topKPolling in combination with the embedding methods mentioned above to encode each private graph into a single vector representation and finally, the system uses a classifier to make the final classification as reliable or unreliable.

During our experiments, we first demonstrate the clear relationship between polarization and misinformation by developing a machine learning classifier that accurately identifies fake news.

Using 5-fold cross validation, our base model using only polarization information can achieve performance of 64% accuracy. These findings clearly reveal a correlation between polarization and the detection of fake news, and we believe it will motivate researchers to look even further into other representations of polarization to help with the challenge of detecting fake news.

Following that, we look at how our framework can be integrated with existing state-of-the-art models to assist in the detection of fake news. We've replicated and experimented with some of the most representative studies in the literature of the most popular methodologies for detecting false news, to examine how adding our work can improve their performance.

Our findings suggest that all of the methods examined in this paper have a considerable performance boost, with the bag of words LSTM model benefiting the most, with a significant 11% improvement, from 66% to 77% accuracy.

1.3. Contributions

In our effort to quantify the relation between polarization and misinformation we collected and labelled a novel Covid19 Fake News Detection dataset. This dataset includes 161,933 articles, 143,070 of which are labelled as reliable and 18,863 are labelled as unreliable. This Dataset is free and open to use, and can be downloaded for from our GitHub Repository.

Moreover, we present in this work four novel Graph based Schemes for modelling Polarization Knowledge. Each Representation able to capture different aspects of polarization

Furthermore, this work describes the first attempt at tackling the Fake News Detection task using only Polarization Knowledge. We achieve this by first constructing a knowledge base of polarization knowledge around a topic in a certain timeframe and embedding it into each article. Finally, we utilize different state-of-the-art ML methods in graphs to classify each article as reliable or unreliable.

The true value of our work can be found when combined with different state of the art approaches in Fake News Detection. After extensive experimentation we found that the addition of our model to every baseline model reconstructed from the literature resulted in a noticeable increase in performance, with RoBERTa based model seeing an increase of 4% accuracy and LSTM based model increasing by an amazing 11% accuracy.

ENDURE can generate and model polarizing knowledge of any subject and across any timeframe in a domain-agnostic manner. Furthermore, it proposes an easy method of applying this knowledge to any downstream task.

We believe that ENDURE's potential to be effortlessly incorporated in downstream tasks can help us develop a better understanding of the phenomenon of polarization.

To summarize, our main contributions are:

- Provide a new Fake News Detection Dataset.
- Provide a multiple novel representation Schemas of polarization information.
- Design the first Fake News Detection Algorithm that uses Polarization Information
- Propose a Framework that can help any approach that aims to solve the fake news detection problem that uses textual information to increase its performance by including Polarization Information
- Extensive Experimentation on most popular Fake news classification approaches, and conclusion on which our framework is most helpful.

The rest of the paper is organized as follows. Section 2 discusses related work in fake news detection and polarization. Section 3 presents the architecture of the system. Section 4 describes first the dataset we collected, then it describes the type of models we used to evaluate our framework, and after that it presents the results of our experiments. Finally, Section 5 concludes this work and discusses possible future work.

Chapter 2

Literature Review

2.1. Structural balance theory

Balance theory is one of the most fundamental and prominent used social theories in the field of social psychology. It dates back to the early seminal work in [16] and is later generalized in [44] having a graph theoretical foundation. It says that signed social networks, meaning graphs with signed edges that indicate friendly/hostile interactions between people, tend to be organized in such a way that conflictual situations are avoided, corresponding to cycles of negative parity.

The theory classifies cycles in a signed network as being either balanced or unbalanced, with a balanced cycle having an even number of negative links and an unbalanced cycle having an odd number of negative links. [54] In essence, it says: “the friend of my friend is my friend” and “the enemy of my friend is my enemy”.

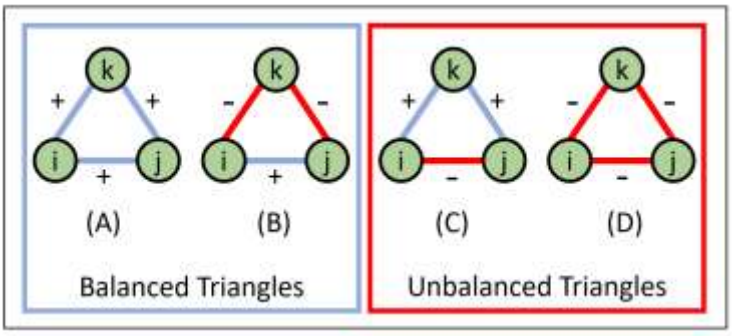


Figure 1: The four undirected signed triangles types according to balance theory. Source [54]

2.2. Fake news detection

There has been a lot of research recently on fake news detection. This work can be divided into four categories [23]:

Knowledge-based methods:

Knowledge-based methods also known as fact-checking, identify fake news by determining if the knowledge contained in the news content is consistent with facts. Pan et al [8] constructed three knowledge graphs using their dataset and a subgraph from an open knowledge graph. They then trained one TransE model for each graph and for each article they calculated the mean probability of each triple existing in each knowledge graph. They then concatenated those probabilities and forwarded them through a classifier.

Han et al [17] reframed the problem of fake news detection as a subgraph classification task. They initially created a knowledge graph from their dataset and then for each article they used a SubGNN to classify the subgraph representing that article.

Style-based methods

Style-based methods are concerned with how the article is written, a common example would be to try to convey extreme emotions. Kolevet al in [13] initially trained a

RoBERTa Model to perform Emotion Classification and then they forwarded the embeddings generated by that model through a Binary Random Forest classifier.

Bhutani et al [9] used tf-IDF vectorizer and tfidf vectorizer with cosine similarity on their dataset and trained a Naive Bayes and Random Forest model to make the classification.

Propagation-based methods:

Propagation-based methods detect fake news based on how it is spread online. Matsumoto et al [10] construct a propagation graph based on how news is shared on Twitter. This graph takes into consideration the different speeds fake and real news spread by using edge weights. It also takes into account the user and textual features that reflect the endogenous preference of each user from the past postings by setting them as node features. Finally, they make the classification using a Graph Transformer Network.

Ren et al [18] constructed a heterogenous graph containing information about each article: The article's topics and subject. These are represented as different node types. They then used a Heterogeneous Graph Neural Network which utilizes an active learning framework to enhance learning performance. The GNN performs node classification on the article nodes.

Source-based methods:

Source-based methods detect fake news by investigating the credibility of news sources at various stages (being created, published online, and spread on social media). Sitaula et al [26] generated a graph where the nodes represent news authors, and the edge between two nodes indicates that the authors have collaborated in writing one or more articles. Authors are categorized as a true authors, fake authors, or an author associated with both true and fake stories. They observed that authors within the same group are more densely connected compared to the authors from different groups.

Multimodal approach:

Paschalides et al [19] created a web browser plugin that uses a combination of source-based, style-based, and Propagation based methods. It first checks whether the domains of the article are in any known fake news domains and fact checks. It then compares a piece of news against known fact-checked articles labelled as fake from Fact-Checking organizations, such as Politifact and Snopes. After that, it analyses user behaviour in social networks and produces a user blacklist of fake news propagators. Finally, it uses a machine learning model which has been trained on linguistic features for the detection of fake news articles.

Mayank et al in [20] use style and content-based methods. For the content-based representation, they find the entities in each article and map them to the open Wikidata Knowledge graph. They then use the ComplEx KG embedding technique to embed the entities. Next, they perform a permutation invariant aggregation of the entities' representation extracted after the KG embedding. For the style-based representation, they use a biLSTM-based neural network to encode the news content. Finally, they concatenate together these two representations and pass them through a classifier.

2.3. Polarization

There are a lot of new works in the literature about polarization, however the majority of them is trying to model it, rather than use it for some classification task.

Myaeng [103] models polarization while trying to identify controversial issues. He uses SentiWordNet [104] for that purpose, a lexical resource in which each WORDNET synset s is associated to three numerical scores Obj(s),Pos(s) and Neg(s), describing how objective, positive, and negative the terms contained in the synset are. He defines the polarity of a term ti to be $\frac{MAX(POS(ti),Neg(ti))}{\sum_{j=1}^{n} MAX(POS(tj),Neg(tj))}$. where n is the number of all the terms in the document.

Mejova et al [105] define polarization as the use of emotional and prejudiced terminology when presenting controversial issues in the news. They capture the expression of sentiment using a series of lexical resources containing words conveying positive and negative emotions.

Guerra et al [21] show that the traditional polarization metric modularity is not a direct measure of antagonism between groups because non-polarized networks can also be divided into relatively modular communities and they propose a new polarization metric based on the analysis of the boundary of a pair of (potentially polarized) communities, that better reflects the notions of antagonism and polarization.

Moreover, the majority of the works in the literature focus on studying primarily the polarization between the Left and Right in the political spectrum. Conover et al [22] demonstrate that the network of political retweets exhibits a highly segregated partisan structure, with next to no connectivity between left and right-leanings. They also show that despite their initial expectations the user-to-user mention network is dominated by a single politically heterogeneous cluster of users in which ideologically opposed people interact at a much higher rate compared to the network of retweets

Anatoliy Gruzd and Jeffrey Roy [106] Investigated Political Polarization on Twitter with a sample of tweets posted during the 2011 Canadian Federal Election. They observed a clustering effect in Twitter around shared political beliefs among supporters of the same party, implying that there exist hotspots of political polarization on Twitter. Furthermore, they discovered evidence of cross-ideological connections and exchanges, which they speculate could allow open, cross-party, and cross-ideological conversation, as well as spark wider debate and learning, as they are viewed by non-affiliated voters and the general media. However, any increased willingness or tendency for committed partisans to shift their allegiances as a result of their Twitter engagements appeared to be far less likely, and they hypothesize that current Twitter usage is more likely to further embed rather than loosen partisan loyalties during election periods, a dynamic that would appear to contribute to political polarization.

Paschalides et al [24] proposed a domain-agnostic and holistic solution, for the identification and measurement of polarizing topics. For that they first identify entities, mentioned in the articles and they calculate the sentiment attitude between a pair of entities, using a lexicon-based classifier of sentence level syntactical dependencies. Next, they cluster these entities into clusters, and identify polarized relationships of this clusters. Finally, they extract topics as clusters of noun phrases that seem to polarize the clusters. *We will be using the output of this framework in the scope of this thesis.*

To the best of our knowledge, this is the first attempt to use polarization information for the task of fake news detection.

Chapter 3

System Architecture

3.1. System Overview

Initially, the user has to provide three essential parameters: the subject, the period of the study being conducted, and a collection of news articles each marked as reliable or unreliable. The subject parameter sets the granularity and scope of the study, the period limits the study in terms of time and the articles are the appropriate dataset on which the framework is going to train on.

After that, ENDURE deploys parallel collectors that fetch the content of news articles matching this criterion. (See section 3.2 Fetch Related Articles Component) Notice that these news articles are different from the initial dataset. To make things clear we will use the terminology "public articles" to refer to the articles collected by our system, and "private articles" to refer to the initial dataset.

Next, the algorithm makes use of the POLAR framework to generate polarization information for the private and public articles. POLAR first constructs the Sentiment Attitude Graph, a signed undirected graph with key entities as nodes and their interactions as edges. It then identifies clusters of entities named Fellowships and polarized relationships between fellowships, called Fellowship Dipoles. Finally, it extracts the discussion topics of Fellowship Dipoles and quantifies each topic's polarization. (see section 3.3 Create Polarization Information Component for more information)

ENDURE then produces a knowledge base of Polarization Information using the data generated from the public articles. This Knowledge base uses one of our proposed graph representation schemas described in section 3.4 Build Graph Representation Component. Each Shema highlights a different aspect of the knowledge and allows the use of different techniques to express this knowledge in the desired downstream task.

Following ENDURE produces node embeddings for the Knowledge Base using of one of the appropriate techniques for the particular schema (see section 3.5 Public Graph Encoding Component). After that we choose a graph representation for our private articles from the selection of schemas mentioned above and embed the knowledge of the of Public Polarization Information into each one using the analogous node embeddings. (see section 3.5 Public Graph Encoding Component)

Afterward, we use topKPolling in combination with the embedding methods mentioned above to encode each private graph into a single vector representation (see section 3.7 Private Graph Encoding component) and finally, the system uses a classifier to make the final classification as reliable or unreliable with the option to use one or more of the other implemented approaches to fake news detection.

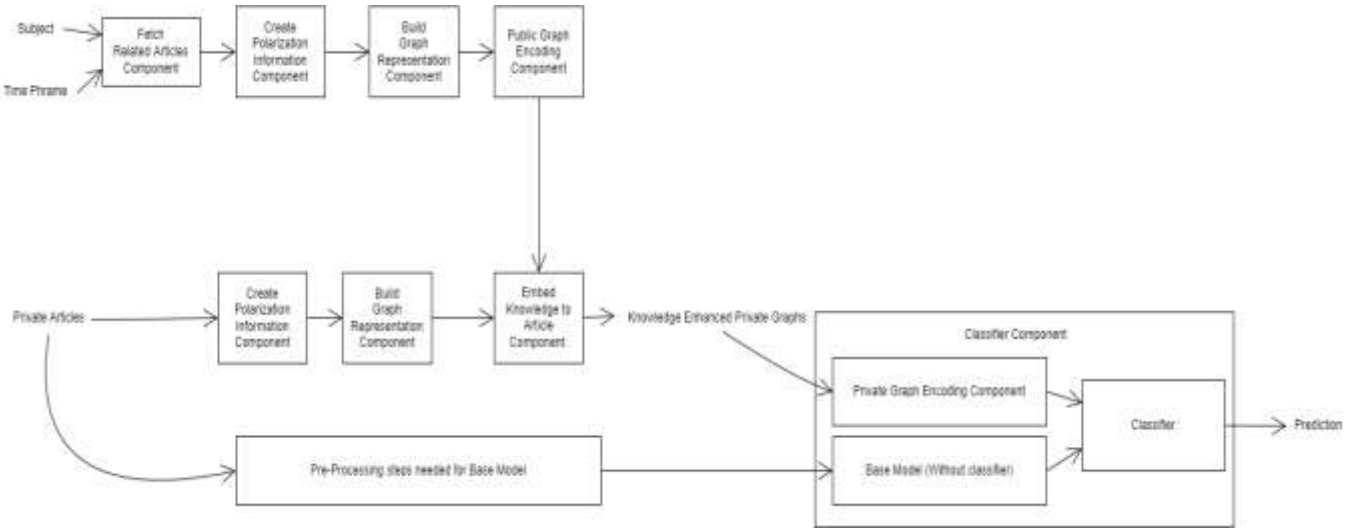


Figure 2: An overview of the proposed framework

3.2. Fetch Related Articles Component

Purpose: This component is responsible for fetching related articles on the given subject and period.

Our framework uses news articles as the primary source of data. The most common approach to modelling polarization in the literature uses data from online social networks, like Facebook or Twitter. [27,28,29,30,31] . The reason online social media is such a popular choice is that it allows easy access to people's opinions on different topics.

These types of data are simply not good enough for our purposes. They are short, noisy, and informal. This makes extracting knowledge exceptionally challenging, often resulting in erroneous and misinterpreted polarization measures. [28]

Our framework chooses to use new articles because, in contrast, they are typically long, descriptive, and formal sources of information. This makes them excellent for processing and extracting knowledge. Furthermore, there is strong evidence In the literature that the current structure of news media (i.e. bias and hyper-partisanship) has a critical role in polarization increase [29,30].

This component requires two user-defined essential parameters, the subject and the period of the study being conducted. We would like here to make a distinction between the subject and the topic. A subject is broader and more general while a topic is more specific. These parameters are used to specify the granularity of the analysis and limit the scope of the study.

After these parameters are specified, it deploys multiple parallel collectors to fetch related news articles from the GDELT Project. The GDELT Project is a large, comprehensive, and open database of global news articles. It monitors the world's broadcast, print, and web news from nearly every country in a multitude of languages and identifies the components driving our global society every second of every day,

creating a free open platform for computing in the entire world. [25]. In addition, the user has the option to manually load a news article dataset for processing as well.

3.3. Create Polarization Information Component

Purpose: This component is responsible for processing the given articles and generating Polarizing information from them. This component relies on the POLAR framework[24].

Overview of the POLAR Framework:

The first step is the Named Entity Recognition and Linking (NERL) where the key article entities are identified and are linked to unique identifiers. It then proceeds to the SAG Construction by identifying the entity pair relationships that serve as edges in the graph. These are based on the co-occurrence frequencies of entity pairs in sentences. When a relationship is identified, then the nature of the relationship (i.e. positive, neutral, or negative) is calculated as the overall sentiment attitude between the entity pair.

Next, it identifies clusters of entities named Fellowships by applying signed network clustering over SAG. It then generates the Fellowship Dipoles, which are the polarization relationships between fellowships. This polarized state of the dipoles is captured using heuristics based on the negative relations across the dipole fellowships, and the structural balance of the dipole

For each of the dipoles, POLAR extracts its discussion topics and quantifies each topic's polarization by extracting noun phrases and clustering them into topics. Lastly, each topic's polarization is quantified by calculating the sentiment attitude of each entity towards the topic's nouns. The topic's polarization is then measured using the polarization index.

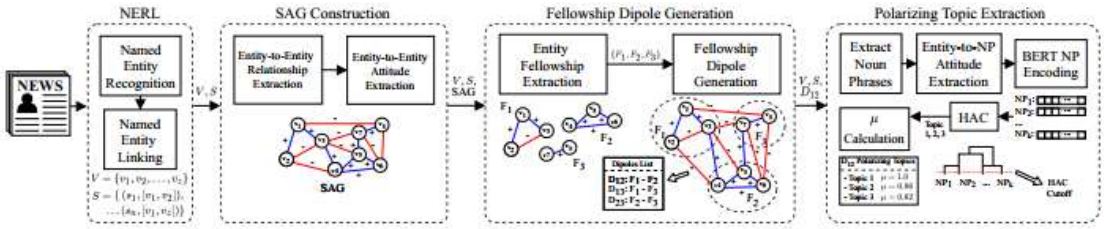


Figure 3: An overview of the POLAR framework.

Following we are going to discuss in detail how each component of Polar works:

Pre-processing:

It is common practice to use some text pre-processing techniques to clean the unstructured text as much as possible and reduce the size of its vocabulary to make it easier for the model to understand. POLAR uses the following text pre-processing techniques:

1. Set everything to lower case:

Replace every uppercase letter, with its lower-case representative

2. Remove stop words:

Stop words are a set of commonly used words in any language. So the removal of stop words is the process of removing the words that are in this set from the text. The reason why the removal of stop words is critical to many applications is that, if we remove the words that are very commonly used in each language, we can focus on the important words instead. [33]

3. Contraction removal:

Contractions are words or combinations of words that are shortened by removing some letters and replacing them with an apostrophe. [32] Contraction removal is the reversal of that process, meaning it removes the apostrophe and it expands the shortened word.

4. Special character and Digit removal:

A special character is a symbol used in writing, that represents something other than a letter or number. A Digit is a character representing a number. Special character and

digit removal is the process where special characters and digits are removed from the text, leaving only letters.

5. Lemmatization:

The goal of lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. It does that by using a vocabulary and performing a full morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma [34]

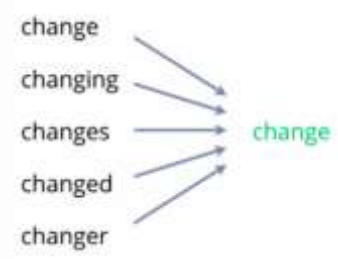


Figure 4: Example of Lemmatization
Source: [92]

6. Tokenization:

Tokenization is the process of breaking the raw text into small chunks. Tokenization breaks the raw text into words or sentences called tokens. These tokens help in interpreting the meaning of the text by analysing the sequence of the words

7. Co-reference resolution:

Coreference resolution is the process of determining linguistic expressions that refer to the same entity in the text.

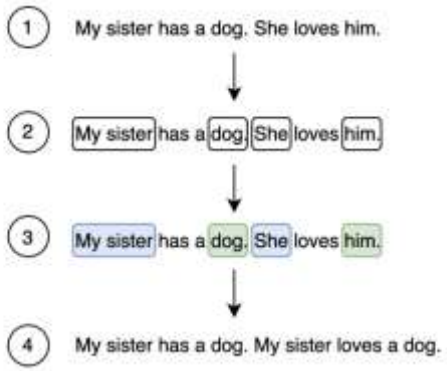


Figure 5: Example of Co-reference Resolution
Source of image: [107]

Identification of Entities

Entities are the foundation of our approach since they populate the social and political groups we analyze. POLAR locates and classifies entities mentioned in the unstructured text of each article into pre-defined types such as person names, organization, location, etc. This is done using a statistical Named Entity Recognition (NER) model able to identify entities within texts as sequences of tokens along with their types

Figure 6: Example of Entity Identification
Source: 108

Named Entity Linking

The problem of performing Named Entity Recognition on a large set of different news articles is that it results in a massive dimensionality, adding noise to the data and making processing computationally expensive. For example, Donald J. Trump can be found in these articles as “Trump,” “President Trump,” and “Donald Trump” are all references to the same person but are treated separately.

Named Entity Linking (NEL) solves this problem by assigning entity mentions to unique identifiers usually with the help of existing Knowledge Graphs (KG) such as Wikipedia. NEL task typically uses 3 steps: First Identify the entity mentions, next find entity candidates and finally apply collective disambiguation. POLAR, uses a snapshot of the Wikidata for this purpose. Wikidata is a free and open knowledge base. It acts as central storage for the structured data of its Wikimedia sibling projects including Wikipedia.[36]

To identify the candidates for an entry mention a string similarity query is performed over the Elastic- search engine, resulting in a small set of possible entities within the Knowledge Graph. The similarity measured use is the similarity measure Token Sort Ratio (TSR) which splits the strings into tokens and then compares them using the simple ratio mechanism, returning 0 if the two strings are completely different and 1 if they are the same. For each entity mentioned, a candidate is considered any knowledge graph node with a TSR score of ≥ 0.5, indicating that at least 50% of the strings are similar.

Following is collective disambiguation over the selected candidate entities. The majority of collective disambiguation approaches in the literature use Machine Learning models trained over textual properties to produce probability scores for the best candidates. Despite these approaches having been shown to have satisfactory performance, they are not suited to POLAR because it is an unsupervised and content-agnostic method, applicable to any knowledge graph.

POLAR uses a domain-agnostic solution to collective disambiguation by encoding the knowledge graph nodes in vectors of low dimensional space, called node

embeddings[37]. Each node is encoded to a d -dimensionality vector representation which captures the characteristics of that node in the structural position within the Knowledge Graph. Moreover, because the processing of knowledge graphs and the training of node embeddings can be done in a distributed manner, this is a more practical solution for large-scale systems. POLAR trains the node embeddings using the DeepWalk algorithm [38], using the suggested training configurations [37]

Finally POLAR can select the best candidate node for each entity mention by searching for semantically similar candidates. Given a tuple of candidates, It identifies how semantically similar they are by calculating the sum of cosine similarities between their node embeddings. POLAR uses a greedy optimization approach recommended by the authors of [37] to reduce the complexity of evaluating all possible candidate combinations.

Figure 7: Example of Named Entity Linking
Source: [94]

Sentiment Attitude Graph Generation

The Sentiment Attitude Graph (SAG) is the basic data model of POLAR. It represents the global entity group, including interactions and attitudes. It is represented as a signed undirected graph with key entities as nodes and their interactions as edges. A signed graph is a graph where each edge has a positive (+1) or negative (-1) sign. [41]

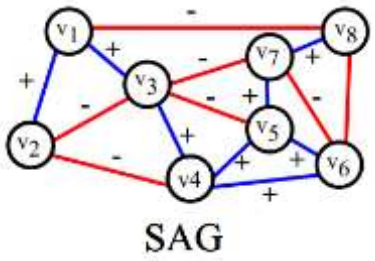


Figure 8: Visual Representation of Sentiment Attitude Graph
Source: [24]

In the previous sections above we have described how the creation of the nodes is implemented. In this section, we describe the construction of the edges.

POLAR first identifies whether there is a relationship between two edges by quantifying their co-occurrences in the news articles. The intuition is that the higher the co-occurrence frequency of an entity pair is, the more probable the existence of a real-life connection between them is.

It populates a binary occurrence matrix with a Boolean value (0 or 1) for sentence si if an entity vj is referred to within the sentence si. The co-occurrence matrix is then calculated by multiplying the dot product of the binary matrix by its transpose. Next, the occurrence matrix is then triangulated, and its diagonal is set to zero to remove the redundant values from being symmetrical. POLAR only keeps the entity pairs whose co-occurrence frequency is over a threshold for more accurate results on the existence of a relationship.

Next POLAR next tries to determine the nature of an entity pair relationship. The nature of an entity-relationship can be described as positive, indicating a possible friendship and supportiveness between the entities, a negative relationship indicating opposition

and hostility, and a neutral relationship indicating the lack of either of these characteristics.

POLAR tries to understand the affection and attitude between an entity pair. This is known as sentiment attitude identification in the literature, where an effort is made to identify the sentiment directed from one element in the text to another. This can be achieved by finding the explicit syntactical dependency path between the entity pair and calculating its sentiment score.

POLAR calculates the sentiment attitude between a pair of entities, using a lexicon-based classifier of sentence-level syntactical dependencies [39]

Given a sentence si, POLAR identifies the entity pair (vx; vy), where vx is the attitude holder, and vy is the attitude target, by extracting all the possible entity pairs. Afterward, it calculates the sentiment attitude from the holder vx towards the target vy, which is denoted as att(si; vx; vy) ∈ {positive, neutral, negative}.

As features, it considers all the syntactical dependency paths between the head word of vx and v y in sentence si. These features include:

i) The sentiment label of the path that contains the dependencies between the subject nsubj and direct object dobj of the sentence si

ii) The sentiment label of the path containing the dependency pattern of a clausal complement ccomp of the subject (nsubj; ccomp; nsubj) of si

iii) An indicator of the negative nominal modifiers dependency nmod: against between the two entities within si.

Taking into account that SAG is an undirected graph, POLAR considers bidirectional relationships. Thus, for each entity pair, it calculates both att(si; vx; vy) and att(si; vy; vx). To calculate the sentiment label, it uses the IBM Debater Sentiment Composition Lexicon [40], because it can better capture the nature of relationships between entities in the concepts of conflicts and debates. After processing the news articles, and collecting the sentiment attitudes for each entity pair, it calculates the average sentiment attitude wxy and populates the edges (vx; vy; wxy) of SAG.

Extraction of Entity Fellowships

Fellowships are clusters of entities characterized by the common beliefs, ideologies, and general supportiveness of their members. Within the SAG this is analogous to densely connected graph partitions with positive attitudes.

Clustering signed networks is the process of finding clusters such that most edges within clusters are positive, and most edges across clusters are negative.
Despite being a relatively new research topic, signed network clustering has several notable papers that use methods based on correlation clustering, k-balanced social theory, and signed modularity. (Tang et al., 2016).

These methods are constrained by their reliance on modularity, which has been demonstrated to have a resolution limit, rendering them incapable of detecting small groups [43]. Small communities must not be overlooked because they may represent significant minorities. Furthermore, these techniques require a certain number of clusters, which is undesirable because POLAR creates a SAG of arbitrary size.

POLAR employs the SiMap approach for identifying fellowships inside SAG to overcome the aforementioned restrictions [42]. SiMap is an extension of the Constant Potts Model (CPM) that is applicable on signed networks. SiMap can partition the SAG into any number of dense positive clusters, which we refer to as fellowships. The resolution is the only configurable parameter instead of the number of clusters. By increasing the resolution, we may see smaller and denser groups in the network, bypassing the resolution limit of modularity-based methods. As suggested by [42], POLAR sets the resolution to be 0.05.

Generation of Polarized Dipoles

By taking every conceivable pair of fellowships, POLAR creates an initial set of fellowship dipoles. However, not all of the dipoles produced are polarized. It uses two heuristic principles to filter out non-polarized dipoles: negative across and frustration.

The first evaluates the ratio of negative edges connecting the dipole's two sides, while the second considers the dipole's structural balance [44] as measured by the frustration index [45]. Polar first applies the negative across heuristic and then the frustration heuristic to reduce the number of dipoles.

The **negative across of two fellowships** is the ratio of negative edges between them. The intuition is that dipoles with a higher negative across, are more likely to be polarized. Polar sets the threshold for this to be 0.5 as it offers the best results while accounting for possible errors.

The **frustration heuristic** utilizes a dipole's frustration index [46], which indicates the distance from total structural balance [44] between two fellowships. A signed graph is said to be balanced if either of the following is true:

i) all the edges are positive
ii) the nodes can be partitioned into two disjoint sets such that positive edges exist only within clusters, and negative edges are only present across clusters.

According to balance theory, social tension and polarization result from balanced structural configurations of entities with signed relations [45]. As a result, a perfectly balanced signed graph can be segregated into two completely opposing and conflicting fellowships. As a result, a fellowship dipole with a high structural balance suggests increased fellowship opposition and a strongly polarized state [45].

POLAR measures structural balance using the frustration index [46], which shows the minimal number of edges in graph G whose removal results in structural balance [46]. As a result, dipoles with higher frustration index have a higher chance of being polarized It construct the normalized frustration index for each dipole, which ranges from 0 to 1, with 0 being completely imbalanced and 1 being perfectly balanced.

POLAR maximizes the number of polarized dipoles by removing the dipoles with a frustration index of less than 0.7

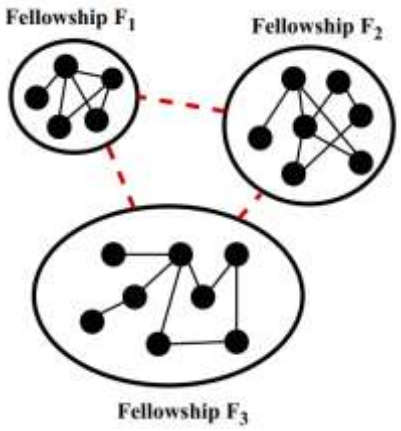


Figure 9: Visual Representation of Fellowships and Fellowship Dipoles.
Source: [24]

Extraction of Polarizing Topics

Given a polarized fellowship dipole, POLAR identifies the discussion topics between the opposing fellowships and measures the polarization around them by processing the sentences where fellowship dipole entities co-occur.

POLAR defines topics as clusters of Noun Phrases (NP) within those sentences. To determine if a topic is polarizing, it calculates its polarization index, a metric proposed by Morales et al in [28] that considers overall attitude observations on a specific topic.

Find noun phrases

Grammatically a Noun Phrase functions as a noun in a sentence and is fundamental in a variety of NLP tasks. Constituency parsing, the task of separating a text into sub-phases or constituents, is one way to identify the noun phrases of a sentence.
POLAR generates and traverses the constituency tree of each sentence in a dipole, collecting the base noun phrases as the leaves of each tree branch labelled as Noun Phrases

Cluster noun phrases into topics

Topics are formed by clustering the noun phrases into groups with similar semantic meanings. To do so, POLAR first reduce the noun phrases' lexical dimensionality with a series of pre-processing steps mentioned above.
To semantically cluster the noun phrases it encodes them into word vectors as they have shown to efficiently and effectively capture the semantic meaning of text [48]. The encoding is done using the novel context-dependent BERT embeddings [49]. Each noun phrase is represented with a 1024-dimensional vector. After the encoding of the word vectors, clustering is applied.

POLAR extracts an arbitrary size of noun phrases per dipole, thus similar to the generation o fellowships, k-clustering techniques are not an option. To this end, POLAR uses the Hierarchical Agglomerative Clustering (HAC) method. In addition, HAC generates a comprehensible hierarchical dendrogram that depicts the interdependencies between topical clusters.

POLAR uses cosine distance as the distance metric, which is recommended when working with word vectors. As a consequence, HAC generates a cluster hierarchy that represents the dipole's discussion topics. POLAR uses a cosine distance threshold of 0.2 to find the final set of topics. This results in a set of discussion topics represented as noun phrases clusters.

Measuring Topic Polarization:
To quantify the topic polarization of a dipole POLAR must first determine the population of sentiment attitudes towards the topic. The attitude population is equal to the attitude instances indicated by dipole fellowship entities towards the topic's noun phrases in each fellowship in the dipole.

These attitudes are determined using an adaptation of the sentiment attitude approach described above. POLAR defines a target noun phrase instead of a target entity, which is denoted within the applied dipole sentence. This is accomplished by examining each entity and available noun phrase pair in the text. POLAR calculates the sentiment attitudes of that pair if there is a dependency path between the node and the Noun phrase.

POLAR proceeds to quantify the topic polarization using the polarization index measurement once it has the sentiment opinions regarding the topics. [25] The intuition for this metric is the following: “a population is perfectly polarized when divided into two groups of the same size and with opposite opinions.” This is akin to a situation in which two people have different attitudes, and the degree of polarization is determined by how drastically opposed their viewpoints are (i.e. the distance among the attitudes) After computing the polarization index for each dipole's topic, POLAR generates a list of polarizing topics, ranked from most polarizing to least polarizing.

In conclusion, the output of this component is the following:

1. Sentiment Attitude Graph (SAG)
   - ✓ A Signed underacted Graph with key entities as vertices (e.g. political figures, organizations, countries, etc.), and their interactions (e.g. supportiveness or opposition) as edges.

2. Fellowships
   - ✓ Clusters of entities that indicate similar feelings towards other Entities/Fellowships

3. Fellowship dipoles
   - ✓ Polarized relationships between Fellowships

4. Discussion topics
   - ✓ Discussion topics that seem to polarize Fellowships

The user is then able to decide which of the above polarization information to generate and use for its representation. Notice that each piece of information requires the generation of the previous.

3.4. Build Graph Representation Component

Purpose: This component takes the output of the Create Polarization Knowledge Component and transforms it into a graph representation of this data.

We have identified and explored four different representations Shemas of this information during our experiments:

Heterogeneous Graph with Edge Features:

A Heterogeneous Graph is a special type of graph in which in addition to having a set of nodes and a set of edges, two mapping functions map each node to a node type and each edge into an edge type [51]. Our representation of a heterogeneous graph contains all the generated information from the aforementioned component in the following format:

Different node types:

1. Entity
- These nodes represent the key entities generated by the articles (political figures, organizations, countries, etc.)
  see section Create Polarization Information Component - Identification of Entities subsection
2. Fellowships
- These nodes represent Clusters of Entities see section Create Polarization Information Component - Extraction of Entity Fellowships
3. Fellowship Dipole (Polarized relationship of two fellowships)
- These nodes represent the polarized relationships between Fellowships (see section Create Polarization Information Component - Generation of Polarized Dipoles). We choose to represent this relationship as a node instead of an edge type between Fellowships because we found that our models are better able to capture the desired information.
4. Topic
- These nodes represent the Discussion topics that seem to polarize Fellowships see section Create Polarization Information Component - Extraction of Polarizing Topics

Edge types:

1. Attitude towards
    - This edge type represents the attitude of an Entity towards another Entity.
    - Edge features:

- Sign: The categorized sentiment of the tail towards the head, (-1 or 1) where 1 indicates supportiveness and -1 indicates opposition.
- Sentiment: The sentiment of the tail towards the head, in the range [-1.1] where 1 means strong supportiveness, while -1 means strong opposition.
- Frequency: The number of sentences these nodes refers to each other, integer

2. Sentiment toward the topic
   - This edge type represents the sentiment of the tail towards the head topic.
   - Edge features:
     - Sentiment: The sentiment of the tail towards the head, in the range [-1.1] where 1 means strong supportiveness, while -1 means strong opposition.
3. Member of
   - This edge type represents that the tail entity is a member of the head Fellowship.
4. Part of
   - This edge type represents that the tail Fellowship is a part of the head Fellowship dipole.
   - Edge features:
     - Frustration index: minimal number of edges in a graph whose removal results in structural balance [0.7,1] where 1 is perfectly balanced, while 0.7 indicates less balanced. The more balanced it is the higher chance of being polarized.
     - Positive edges: number of positive edges across conflicting fellowships, integer
     - Negative edges: number of negative edges across conflicting fellowships, integer
     - Positive ratio: percentage of positive edges across conflicting fellowships
     - Negative ratio: percentage of negative edges across conflicting fellowships
5. Discussion of
   - This edge type represents that the tail Fellowship Dipole discusses and is polarized by the head topic.
   - Edge features:
     - Polarization: The polarization index that occurs between the tail fellowship Dipole because of their sentiments toward the head topic. [0,1] where 1 means strong polarization, while 0 means no polarization.

This is the natural way of representing the generated information because it explicitly shows how different types of polarization information relate to each other.

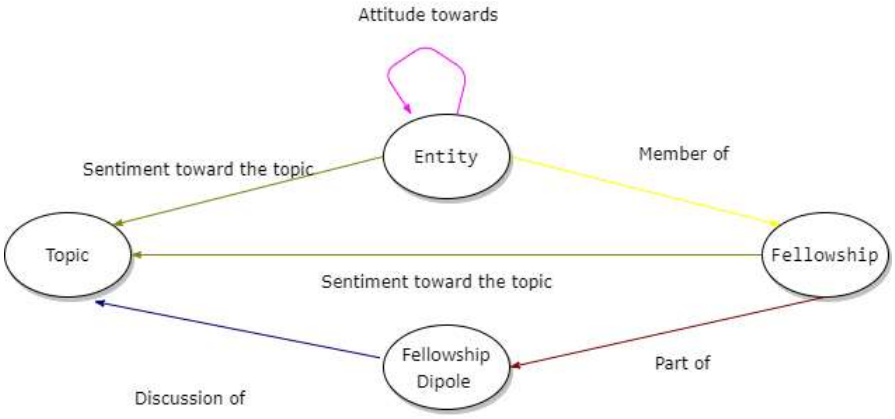


Figure 10: Heterogeneous Graph with Edge Features Shema

Heterogeneous Graph without Edge Features:

Because there are some techniques that cannot utilize edge features, we came up with an alternative representation that doesn't make use of them. In our best effort to keep as much useful information as possible we model the previous edge features as either a different node with them being node features or as different edge types. So, the second representation is the following:

Different node types:

1. Entity
- These nodes represent the key entities generated by the articles (political figures, organizations, countries, etc.)
  see section Create Polarization Information Component - Identification of Entities subsection
2. Fellowships
- These nodes represent Clusters of Entities see section Create Polarization Information Component - Extraction of Entity Fellowships
3. Fellowships Dipole (Polarized relationship of two fellowships)
- These nodes represent the polarized relationships between Fellowships (see section Create Polarization Information Component - Generation of Polarized Dipoles). We choose to represent this relationship as a node instead of an edge type between Fellowships because we found that our models are better able to capture the desired information.
4. Topic
- These nodes represent the Discussion topics that seem to polarize Fellowships see section Create Polarization Information Component - Extraction of Polarizing Topics
5. Polarization
- These nodes represent the amount of polarization that occurs between the fellowship dipole (which that points to it) because of their sentiments toward the topic (which this node points to) as measured by the polarization index .
6. Frustration
- These nodes represent the frustration index that occurs between the fellowships of the dipole that points to it.

Note that the two new node types mentioned above represent a continuous range of values for the appropriate metric.

Edge types:

1. Opposition towards
   - This edge type represents the opposition of the tail towards the head.
2. Support towards
   - This edge type represents the support of the tail towards the head.

   Member of
   - This edge type represents that the tail entity is a member of the head Fellowship.
3. Is polarized
   - This edge type represents that the tail Fellowship Dipole is polarized by the amount represented by the head polarization node.
4. Causes polarization
   - This edge type represents that the head topic causes polarization specified by the amount represented by the tail polarization node.
5. Has frustration index
   - This edge type represents that the tail Fellowship has a frustration index indicated by the amount represented by the head frustration index node.
6. Causes frustration index
   - This edge type represents that the head Fellowship Dipole causes frustration index specified by the amount represented by the tail frustration index node.

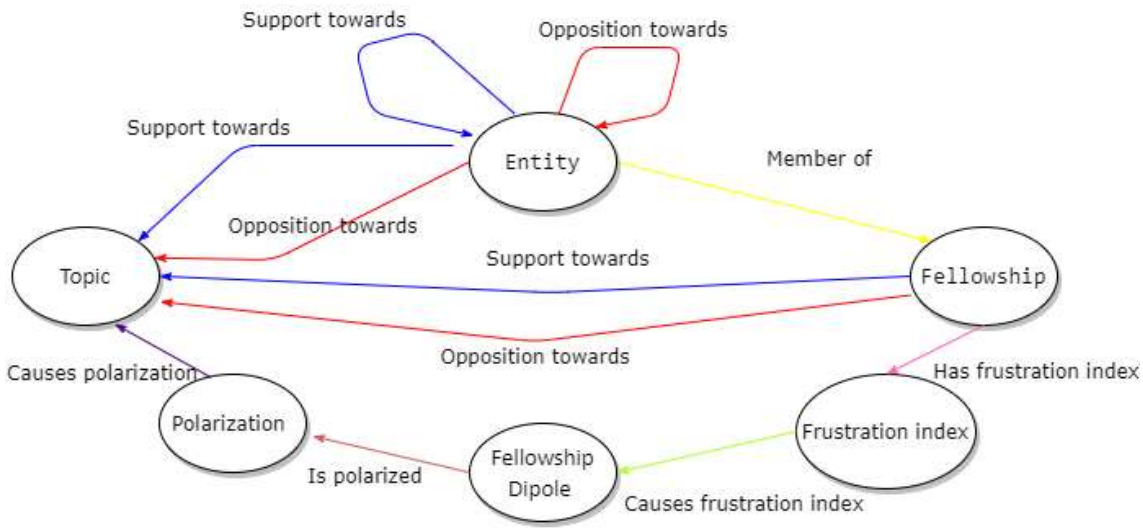


Figure 11: Heterogeneous Graph without Edge Features Shema

Knowledge Graph:

A knowledge graph is a directed labelled graph in which the labels have well-defined meanings. A directed labelled graph consists of nodes, edges, and labels where an edge connects a pair of nodes and captures the meaning of that relationship with a label. More formally, given a set of nodes N, and a set of labels L, a knowledge graph is a subset of the cross-product N × L × N. Each member of this set is referred to as a triple (subject, predicate, object) and can be visualized in figure 12 [52]

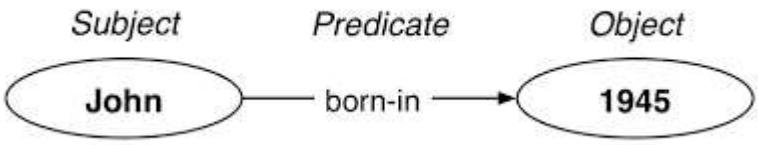


Figure 12: Example of a Knowledge Graph Triple
source [109]

Knowledge Graphs are a very exciting and increasingly popular representation of knowledge in the literature, with recent advancements in knowledge graph embeddings (KGE) being especially promising. However, it offers a less expressive data model from the Heterogeneous Graph. To overcome these limitations, we need to use a different definition of the knowledge generated by the POLAR.

The major limitation of this data model (when using open-source libraries) is that we are not able to use continuous values, thus we lose the information represented as edge features in the heterogeneous graph representation. To overcome this, we put the most important values into buckets representing a range of values and use this instead. Furthermore, there is no way of using node type, so we do that using a label

For our representation of a knowledge graph, we used the following labels:

1. Support
    - This label indicates that subject fosters positive sentiments/ supports the object

- This label represents the " Attitude towards" and "Sentiment towards the topic" edges that indicated strong supportiveness

2. Oppose
   - This label indicates that subject fosters negative sentiments/ opposes the object
   - This label represents the " Attitude towards" and "Sentiment towards the topic" edges that indicated strong opposition
3. Member of
   - This label indicates that the subject (Entity) is a member of the object(Fellowship)
4. Part of
   - This label indicates that the subject (Fellowship) is part of the object(Fellowship Dipole)
5. Low Polarization
   - This label indicates that there is relatively low polarization between the subject Fellowship Dipole because of their sentiments toward the object topic
   - This label represents the "Discussion of" edges that indicated relatively low Polarization
6. Medium Polarization
   - This label indicates that there is relatively medium polarization between the subject Fellowship Dipole because of their sentiments toward the object topic
   - This label represents the "Discussion of" edges that indicated relatively medium Polarization
7. High Polarization
   - This label indicates that there is relatively high polarization between the subject Fellowship Dipole because of their sentiments toward the object topic
   - This label represents the "Discussion of" edges that indicated relatively high Polarization
8. Type of
   - This label indicates what kind of knowledge represents each node $\in$ {Entity, Fellowship, Fellowship Dipole, Topic}

Notice that despite our best efforts, we are still not able to include all the information represented by the previous representation, some can be represented implicitly in the labels, and other less important information is not included at all.

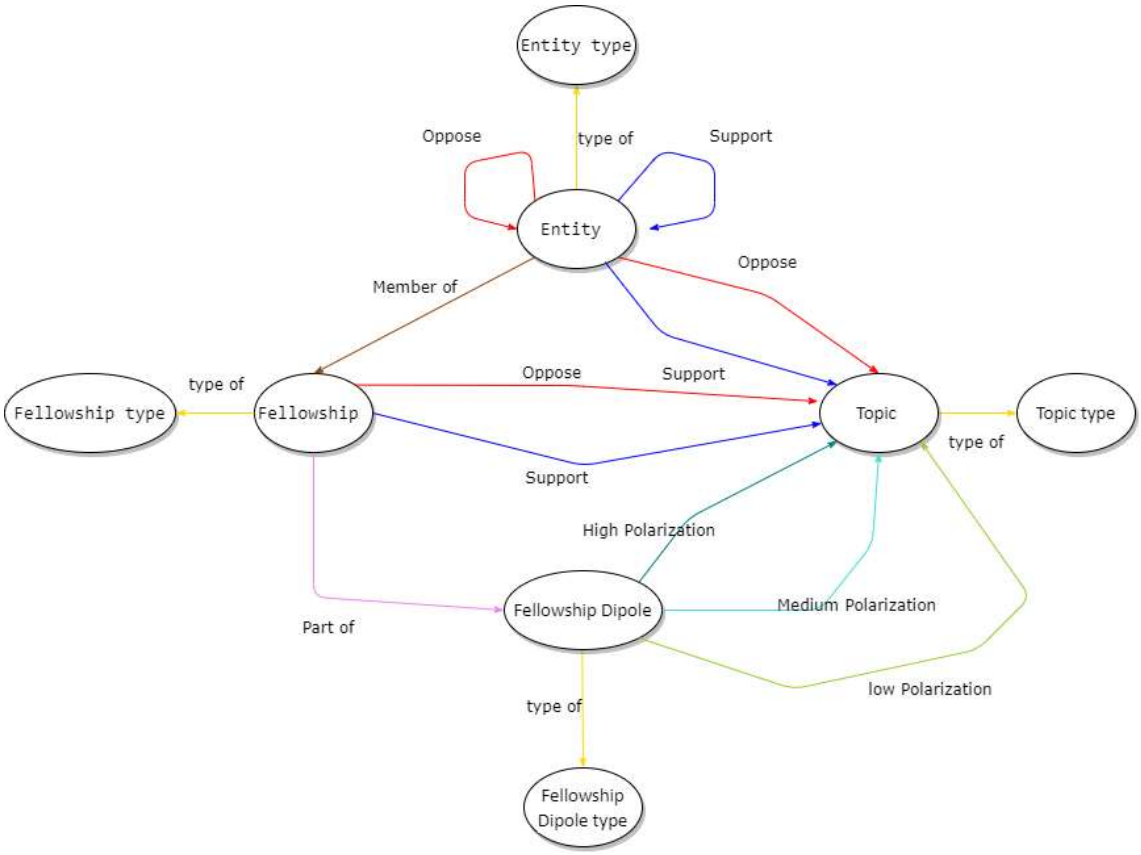


Figure 13: Knowledge graph Shema

Sentiment Attitude Graph (SAG)

This is a signed undirected Graph with key entities as nodes (e.g. political figures, organizations, countries, etc.), and their interactions (e.g. supportiveness or opposition) as edges. A signed graph is a graph where each edge has a positive (+1) or negative (-1) sign.

We wanted to experiment with using just including the SAG, which all other information is based on because we would be able to use a rawer format to allow deep neural networks to find the optimal features for this problem[53]. Moreover, we allow the use of state-of-the-artwork signed networks (described below) and incorporate social theories. We are particularly interested in particular in balance theory.

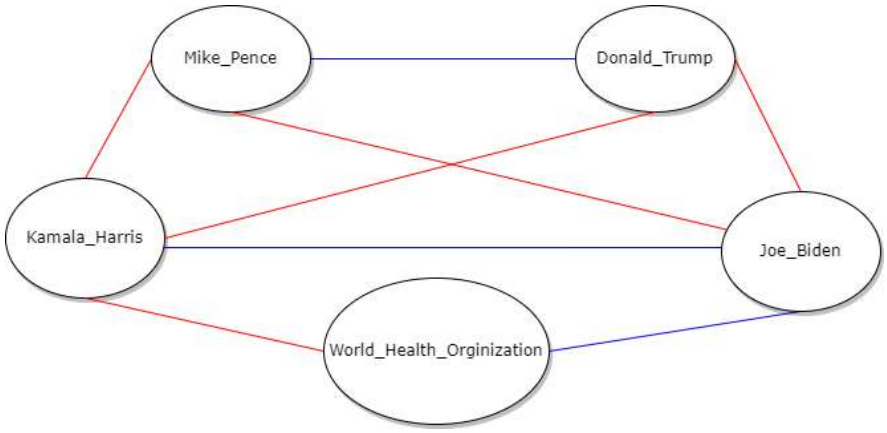


Figure 14: Example of Sentiment Attitude Graph

3.5. Public Graph Encoding Component

Purpose This component is responsible for creating node embeddings. This means mapping the nodes and their relations within a graph into a low-dimensional vector for each node, whose geometric relationships in the embedding space reflect the structure of the original graph [56].

The challenge with graph encoding is that there is no straightforward way to encode this high-dimensional non-Euclidean graph structural information into a feature vector.[67]

The Encoder-decoder framework

Hamilton et al [67] developed a unified encoder-decoder framework, which explicitly organise the various node embeddings methods around two key mapping functions: an encoder, which maps each node to a low-dimensional vector, or embedding, and a decoder, which decodes structural information about the graph from the learned embeddings.

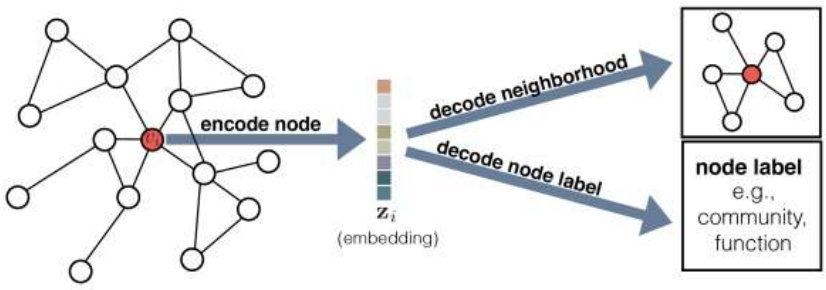


Figure 3: Overview of the encoder-decoder approach. First the encoder maps the node, v_i, to a low-dimensional vector embedding, $\mathbf{z}_i$, based on the node's position in the graph, its local neighborhood structure, and/or its attributes. Next, the decoder extracts user-specified information from the low-dimensional embedding; this might be information about v_i's local graph neighborhood (*e.g.*, the identity of its neighbors) or a classification label associated with v_i (*e.g.*, a community label). By jointly optimizing the encoder and decoder, the system learns to compress information about graph structure into the low-dimensional embedding space.

Source: [67]

The idea behind the encoder-decoder approach is that if we can learn to decode high-dimensional graph information from encoded low-dimensional embeddings, such as the global positions of nodes in the graph or the structure of local graph neighborhoods, then these embeddings should, in theory, contain all the information needed for downstream machine learning tasks.

Formally, the encoder is a function that maps nodes to vector embeddings, and the decoder is a function that takes a set of node embeddings and extracts user-specified graph statistics from them. The pairwise decoder reconstructs the similarity between two embeddings in the original graph, and the goal is to optimize the encoder and decoder mappings to minimize the error, or loss, in this reconstruction. [67]

Once the encoder-decoder system has been optimized, the trained encoder can then be used to generate embeddings for nodes, which in turn can be used as feature inputs for downstream machine learning tasks. [67]

By Adopting this encoder-decoder view, we can organize our discussion of the following node embedding methods along with the following four methodological components: [67]

1. A pairwise similarity function, defined over the graph.
    - This function measures the similarity between nodes in the graph
2. An encoder function (ENC) generates the node embeddings.
    - This function contains several trainable parameters that are optimized during the training phase.
3. A decoder function (DEC), which reconstructs pairwise similarity values from the generated embeddings.
    - This function usually contains no trainable parameters.
4. A loss function, which determines how the quality of the pairwise reconstructions is evaluated to train the model.
    - i.e., how the decoder output is compared to the true similarity between the nodes.

Encoding the Heterogeneous Graph:

The encoding techniques can be roughly categorized into shallow and deep embedding approaches.

Shallow embedding approaches

In shallow embedding approaches, the encoder function is simply a lookup table. The embeddings are a $R^{d\times|V|}$ matrix containing the embeddings vectors for all nodes. The set of trainable parameters is simply this matrix.

We will not focus on shallow embeddings techniques because they are becoming increasingly rare in the literature, however for the sake of completeness we are going to include the most widely used shallow embedding technique, node2Vec.

Node2Vec optimizes embeddings to encode random walk statistics. Random walks is a technique for optimizing node embeddings so that nodes that tend to co-occur on short random walks over the graph have similar embeddings.

A walk is a sequence of nodes ($w0$,$w1$, . . .wt) where ($w\tau$,$w\tau$+1) ∈ E. The probability of the walk is the product of stepwise transitions:

$$\begin{aligned}\text{Prob}(l) &= \prod_{(w_\tau, w_{\tau+1})\in l} \text{Prob}(w_{\tau+1}|w_\tau) \\ &= \prod_{(w_\tau, w_{\tau+1})\in l} |A|_{w_\tau w_{\tau+1}}/d_{w_\tau}\end{aligned}$$

The t-step random-walk transition probability from node u to v can be expressed as the sum of probabilities of all length-t walks between u and v, denoted as Walk(u, v, t):

$$|M|_{uv}(t) = \sum_{l\in \text{Walk}(u,v;t)} \text{Prob}(l)$$

which serves as a measure of topological similarity between u and v. The Markov time t controls the scale of the walk. [75]

The random walk in Node2vec is biased by two random walk hyperparameters, p, and q. The hyperparameter p controls the likelihood that the walk will immediately return to a node, whereas q controls the likelihood that the walk will return to a node's one-hop neighbourhood.

These hyper-parameters allow the model to choose the degree to which learning embeddings focus on community structures and local structural roles.
Furthermore, Node2Vec uses a decoder based on the inner product

$$\begin{aligned}\text{DEC}(\mathbf{z}_i, \mathbf{z}_j) &\triangleq \frac{e^{\mathbf{z}_i^\top \mathbf{z}_j}}{\sum_{v_k\in\mathcal{V}} e^{\mathbf{z}_i^\top \mathbf{z}_k}} \\ &\approx p_{\mathcal{G},T}(v_j|v_i),\end{aligned}$$

where $p_{G,T}(v_j|v_i)$ is the probability of visiting v_j on a length-T random walk starting at v_i, with T usually defined to be in the range $T \in \{2, ..., 10\}$. Note that this similarity measure is both stochastic and asymmetric. The loss function it tries to minimize is the following cross-entropy loss:

$$\mathcal{L} = \sum_{(v_i,v_j)\in\mathcal{D}} -\log(\text{DEC}(\mathbf{z}_i, \mathbf{z}_j)),$$

where the training set D is generated by sampling random walks starting from each node.

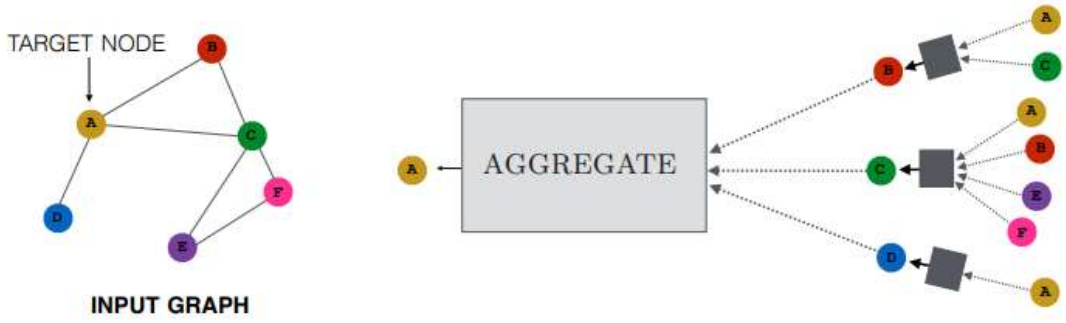
Because naively evaluating the loss above is prohibitively expensive node2vec approximates it using negative sampling. instead of normalizing over the full vertex set, node2vec approximates the normalizing factor using a set of random "negative samples". (More on negative samples in section Encoding the Knowledge Graph)

Graph Neural Networks

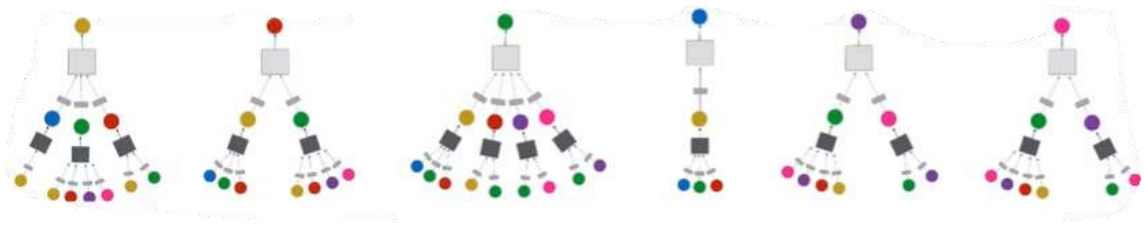
Deep embedding approaches use more complex encoders which depend more generally on the structure and attributes of the graph. [67] We have decided to focus on Graph Neural Networks techniques for this section because they are by far the most popular and promising method in the literature.

For this representation, we have decided to use the most influential Graph Neural Networks including Graph Convolutional Networks(GCN), GraphSAGE, and Graph Attention Networks (GAT)

GNNs are deep learning models aiming at addressing graph-related tasks in an end-to-end manner [68]. GNNs produce embeddings for a node by aggregating information from its local surroundings. Because they represent a node as a function of its surrounding neighbourhood in a way analogous to the receptive field of a centre-surround convolutional kernel in computer vision, these aggregations are commonly referred to as convolutional.



The neighbourhood aggregation methods iteratively aggregate the representation for a node throughout the encoding step. First, the node embeddings are set to the same values as the input node attributes.



Next, during each iteration of the encoder algorithm, nodes accumulate inputs from their neighbors based on the following formula:

$$\mathbf{h}_i^k = \sum_{v_j \in \mathcal{N}(v_i)} h(\mathbf{h}_j, \mathbf{x}_i, \mathbf{x}_j),$$

Where h is an arbitrary differentiable function of the form h (R^d X R^m X R^m à R^d). For the specified number of epochs, this equation is applied recursively. Using an aggregation function that operates across sets of vectors, nodes aggerate the embeddings of their neighbours after each iteration.

Following this aggregation, each node is given a new embedding that is equal to the sum of its aggregated neighbourhood vector and its prior embedding from the previous iteration.

Finally, the procedure is repeated by feeding the aggregated embedding via a thick neural network layer.

The node embeddings comprise information aggregated from further and further reaches of the graph as the process iterates. Yet, the dimensionality of the embeddings stays constrained, forcing the encoder to compress all of the neighborhood information into a single allow-dimensional vector.

The process ends after the specified number of iterations, and the resulting embedding vectors are the output as node representations.
Unlike shallow embedding techniques, this method's trainable parameters are shared across nodes.

To generate embeddings for all nodes, the same aggregation function and weight matrices are utilized, and only the input node attributes and neighborhood structure differ depending on which node is being embedded.

This parameter sharing improves efficiency (i.e., parameter dimensions are independent of graph size), offers regularization, and allows this method to construct embeddings for nodes that were not observed during training.

GCN uses this framework, with the update rule being the following:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

Source: [110]

Where H is the feature matrix and W is the trainable weight matrix. When we look at it from an individual node perspective, the update rule can is the following:

$$h_i^{(l)} = \sigma(\sum_{i \in N_j} \frac{1}{\sqrt{|Ni||Nj|}} W h_j)$$

Where Ni and Nj are the sizes of the nodes' neighbourhoods. [72]

The GCN has the coefficient indicated by the figure below which is multiplied in our projection of the node features.

$$\frac{1}{\sqrt{|N_i||N_j|}}$$

This coefficient is derived from the graph's degree matrix and is highly dependent on the graph's structure. It indicates how significant the node's j attributes are for node i.

The main idea of the second GNN we are going to utilize, GAT, is to compute that coefficient implicitly rather than explicitly, as GCNs do. As a result, we may utilize more information than just the graph structure to determine the importance of each node. This is accomplished by treating the coefficient as a learnable attention mechanism.

The attention mechanism, first proposed by Bahdanau et al in [74], allows the model to use the most relevant sections of the input sequence in a flexible manner by using a weighted combination of all of the encoded input vectors with the most relevant vectors receiving the highest weights. [73]

Velikovi et al. proposed in [70] that the coefficient, which we will refer to as aij, be computed using node attributes and then passed through an attention function. Finally, the softmax function is applied to that result in a probability distribution using the attention weights aij. On a mathematical level, we have the following:

$$a_{ij} = attention(h_i, h_j) \qquad a_{ij} = \frac{exp(a_{ij})}{\sum_{k \in N_i} exp(a_{ik})}$$

Visually it can be represented in the figure below

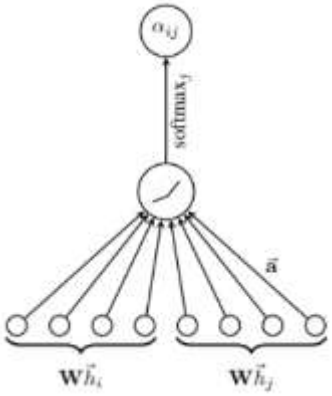


Figure 15: The attention mechanism parametrized by a weight vector a, applying a LeakyReLU activation
Source[70]

So in conclusion, the update rule for GAT for a single node is now the following: [72]

$$h_i^{(l)} = \sigma(\sum_{i \in N_j} a_{ij} W h_j)$$

The above framework explains how to work with homogeneous graphs. It can't be used to heterogeneous graph data trivially since different types of node and edge features can't be processed by the same functions due to feature type inconsistencies. Implementing message and update functions separately for each edge type is a simple technique to get around this. [71]

For the self-supervised training the embeddings generated by these methods are then forwarded into a classifier for each task (node classification, edge classification, edge polarization regression) and the loss is a weighted average for the importance of each task in finding a satisfactory overall representation of the graph.

Encoding the Knowledge Graph:

Knowledge graph embeddings are becoming more and more prominent in the literature. These embeddings are calculated in such a way that they satisfy particular properties, also known as adhering to a KGE model.

These KGE models define different score functions that measure the distance between two entities relative to their relation type In the low-dimensional embedding space. The score functions are then used to train the KGE models so that entities connected by relations are more similar than those that are not.

In this study, we have experimented with the most influential KGE models. KGE models can be roughly categorized into two groups: translational distance models which use distance-based scoring functions and semantic matching models which use similarity-based ones. [64]

Translational distance models

Translational distance models use distance-based scoring functions which quantify the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation. [64]

TransE [57] is by far the most influential KGE model and the most representative translational distance model. It uses vectors to express both entities and relations in the same space. Given a fact (h,r,t), the relation r is interpreted as a translation vector, allowing the embedded entities h and t to be connected with a minimal error by that relation i.e. $h + t \approx r$.

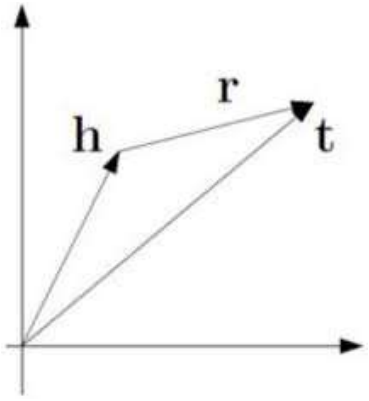


Figure 16: Visualization of TransE
Source: [111]

The scoring function is then defined as the (negative) distance between subject + object and relation:

$$f_r(h,t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}.$$

For a true triple(h,r,t) the resulting score is expected to be large. Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to-N, N-to-1, and N-to-N relations. [64]

To overcome the disadvantages of TransE in dealing with 1-to-N, N-to-1, and N-to-N relations, an effective strategy is to allow an entity to have distinct representations when involved in different relations.

In TransR[58], entities are represented as vectors in an entity space R^d, and each relation is associated with a specific space R^k and modelled as a translation vector in that space.

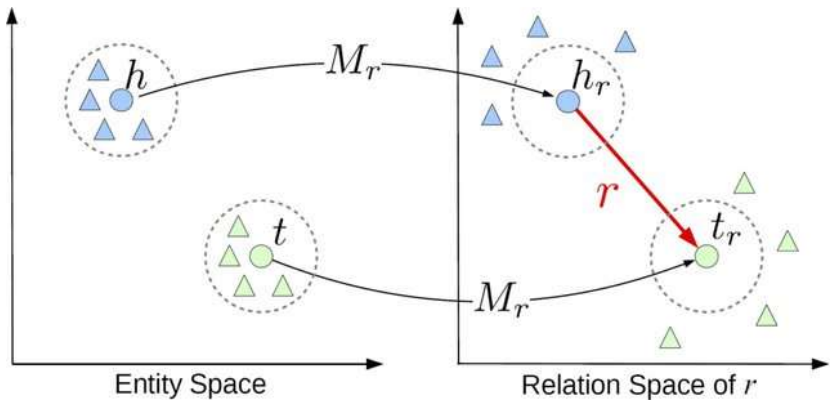


Figure 17: Visualization of TransR
Source: [95]

Given a fact (h,r,t), TransR first projects the entity representations h and t into the space specific to relation r, i.e.,

$$\mathbf{h}_\perp = \mathbf{M}_r\mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r\mathbf{t}.$$

Here $M_r \in R^{kxd}$ is a projection matrix from the entity space to the relation space of r. Then, the scoring function is defined as

$$f_r(h,t) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2.$$

Although powerful in modelling complex relations, TransR introduces a projection matrix for each relation, which requires O(d*k) parameters per relation. So it loses the simplicity and efficiency of TransE (which model relations as vectors and require only O(d) parameters per relation) [64]

KG2E [59] represents entities and relations as random vectors derived from multivariate Gaussian distributions., i.e.,

$$\begin{aligned}\mathbf{h} &\sim \mathcal{N}(\boldsymbol{\mu}_h, \Sigma_h),\\ \mathbf{t} &\sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t),\\ \mathbf{r} &\sim \mathcal{N}(\boldsymbol{\mu}_r, \Sigma_r),\end{aligned}$$

where $\mu_h, \mu_t, \mu_r \in R^d$ are mean vectors, and $\Sigma_h, \Sigma_t, \Sigma_r \in R^{dxd}$ covariance matrices. Following that, KG2E scores a fact by measuring the distance between the two random vectors of t - h and r, which is motivated by the translational assumption, i.e., the two distributions of N (μ_t - μ_h, Σ_t - Σ_h) and N(μ_r Σ_r). There are two different types of distance measures used. One is the Kullback-Leibler divergence [58] which defines:

$$\begin{aligned}f_r(h,t) &= -\int \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_h, \Sigma_t + \Sigma_h) \ln \frac{\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_h, \Sigma_t + \Sigma_h)}{\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_r, \Sigma_r)} d\mathbf{x}\\ &\propto -\mathrm{tr}(\Sigma_r^{-1}(\Sigma_h + \Sigma_t)) - \boldsymbol{\mu}^\top \Sigma_r^{-1} \boldsymbol{\mu} - \ln \frac{\det(\Sigma_r)}{\det(\Sigma_h + \Sigma_t)},\end{aligned}$$

The other one is the inner product probability which defines:

$$\begin{aligned}f_r(h,t) &= \int \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_h, \Sigma_t + \Sigma_h) \cdot \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_r, \Sigma_r) d\mathbf{x}\\ &\propto -\boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu} - \ln(\det(\Sigma)).\end{aligned}$$

Here $\mu = \mu_h + \mu_r - \mu_t$ and $\Sigma = \Sigma_h + \Sigma_r - \Sigma_t$. With the help of Gaussian embeddings, KG2E can effectively model uncertainties of entities and relations in knowledge graphs. [64]

Semantic matching models

Semantic matching models use similarity-based scoring functions. They measure the plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations. [64]

RESCAL [61] maps each entity with a vector to capture its latent semantics. Each relation is represented as a matrix that models pairwise interactions between latent factors. The score of a fact (h,r,t) is defined by the following bilinear function

$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [\mathbf{M}_r]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j,$$

Where h, $t \in R^d$ are vector representations of the entities, and $M_r \in R^{dxd}$ Is a matrix associated with the relation. This score captures pairwise interactions between all h and t components which require $O(d^2)$ parameters per relation[64]

DistMult [62] simplifies RESCAL by restricting M_r to diagonal matrices. It introduces a vector embedding $r \in R^d$ for each relation r, and requires M_r = diag(r). Thus the scoring function is defined as:

$$f_r(h,t) = \mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \mathbf{t} = \sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_i.$$

The above score captures pairwise interactions between only h and t components along the same dimension. It also reduces the number of parameters to O(d) per relation. However, since h^T * diag(r) * t = t^T * diag(r) * h for any h and t, this over-simplified model cannot deal with non-symmetric relations, thus making it insufficient for general Knowledge graphs. [64]

ComplEx [63] extends DistMult by introducing complex-valued embeddings to better model asymmetric relations. In ComplEx, entity and relation embeddings h,r,t no longer lies in real space butin a complex space. The score of a fact (h,r,t) with this model is defined as:

$$f_r(h,t) = \mathrm{Re}\left(\mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \bar{\mathbf{t}}\right) = \mathrm{Re}\left(\sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\bar{\mathbf{t}}]_i\right),$$

where t^{---} is the conjugate of t and Re(.) means taking the real part of a complex value. This scoring function is no longer symmetric, and facts resulting from asymmetric relations may receive different scores depending on the order in which the entities are involved. [64]

Training.

Training can be done under Open World Assumption, which states that knowledge graphs contain only true facts and non-observed facts can be either false or just missing, or under closed world assumption which assumes that all facts that are not contained in the knowledge graph are false. [64]

For our purposes, we choose the open-world assumption since we know that there is missing information that could be true, for example, POLAR clusters users in at most one fellowship, while we know that they could be in more than one. [24]

We can use Negative examples to train our models under the Open World Assumption. In this case, negative examples are triples(h,r,t) that are most likely not true, meaning that they are not in the Knowledge Graph. Negative examples can be generated by replacing the head h or the tail t with a random entity sampled from V or r with a random edge sampled from E.

If you uniformly sample edges and/or relations, you can get false-negative examples. False-negative examples are triples that do exist in the Knowledge Graph. There are several techniques of sampling entity and/or relation while decreasing the probability of false-negative cases to overcome this. For instance, Wang et al in [65] first compute the average number of tail entities per head (tph) and the average number of head entities per tail (hpt). The fact is then corrupted by substituting the head with probability tph / (tph+hpt) and the tail with probability hpt / (tph+htp) for any positive fact from that relation.

Trouillon et al showed in [63] found that the logistic loss is better for semantic matching models like DistMult and ComplEx, whereas the pairwise ranking loss is better for

translational distance models like TransE, TransR, and KG2E, thus we chose those as examples.

The logistic loss:

$$\min_{\Theta} \sum_{\tau \in \mathbb{D}^+ \cup \mathbb{D}^-} \log\left(1 + \exp(-y_{hrt} \cdot f_r(h,t))\right)$$

Source[64]

where t =(h,r,t) is a training example in true {true facts U negative examples} and y_{hrt} = 1 if it is a positive example (true fact) or 0 if it is a negative example. Minimizing the logistic loss, as demonstrated by Bouchard et al in [66], can aid in the discovery of compact representations for some complex relational patterns, such as transitive relations.

Pairwise ranking loss:

$$\min_{\Theta} \sum_{\tau^+ \in \mathbb{D}^+} \sum_{\tau^- \in \mathbb{D}^-} \max(0, \gamma - f_r(h,t) + f_{r'}(h',t'))$$

Source[64]

Pairwise ranking loss makes the scores of positive facts higher than those of negative ones. Here, t^+ = (h,r,t) is a positive example, t^-= (h',r',t') is a negative example and γ is a margin separating them. Minimizing the pairwise ranking loss has another benefit. It does not imply that negative instances are always wrong, merely that they are more invalid than positive examples.

Encoding the Sentiment Attitude Graph (SAG):

As mentioned above the SAG is basically a signed undirected Graph with key entities as nodes (e.g. political figures, organizations, countries, etc.), and their interactions (e.g. supportiveness or opposition) as edges.

For the encoding of this representation, we wanted to take advantage of two state of the art approaches on signed networks which use the balance theory. We identified the most influential Signed Graph Neural Network, SignedGCN, and POLE which uses signed random walks and further captures the polarization within the graph.

In a nutshell POLE[75] is an embedding method for signed polarized graphs that captures both topological and signed similarities jointly via signed autocovariance and it is based on signed random walks.

We have defined what a random walk is for unsigned graphs in section (3.5 Public Graph Encoding Component in the subsection Encoding the Heterogeneous Graph). For signed graphs POLE defines a signed random walk by continuing to keep track of probabilities of walks for topological similarity and adding an inferred sign for each walk to capture signed similarity.

$$M_{uv}(t) = \sum_{l \in \text{Walk}(u,v;t)} \text{Sign}(l)\,\text{Prob}(l)$$

where Sign(l)determines the sign of the walk l between u and v. POLE uses balancing theory [18] to deduce the sign of the walk, which declares, the famous rule "an enemy of my enemy is my friend." as described above.

It's worth noting that after adding signs to the walks, it's no longer a probability, but rather a notion of signed similarity between u and v. Despite the fact that it is not stochastic, POLE employs a signed random-walk transition matrix called $M(t) \in R^{nXn}$.

The key advantage of POLE's signed random walk is that it guarantees polarized similarity consistency. They define polarized similarity consistency as "Positively related node pairs are more similar than unrelated topologically distant pairs, which are in turn more similar than negatively related pairs."

A signed network is considered polarized by POLE if it has antagonistic communities with dense positive connections within each community and sparse negative connections between them. POLE defines the node-level polarization as the Pearson correlation between a node's signed and unsigned random-walk transitions

$$\mathrm{Pol}(u;t) = \mathrm{corr}(|M|_{:u}(t), M_{:u}(t))$$

and the graph-level polarization as the mean node level polarization for all nodes in the graph:

$$\mathrm{Pol}(G;t) = \underset{u \in G}{\mathrm{mean}}(\mathrm{Pol}(u;t))$$

By changing the Markov time t, the random-walk based polarization suggested in the paper can quantify polarization at different structural scales. A big t measures polarization between macro-level communities (e.g., political party enmity), whereas a small t measure it at the local level (e.g., disagreement between factions within a party).

POLE defines its signed autocovariance based on $M(t)$ as

$$R(t) = M(t)^T W M(t) \qquad W = \frac{1}{\mathrm{vol}(G)} D - \frac{1}{\mathrm{vol}(G)^2} dd^T$$

Where $W \in \mathrm{R}^{n \times n}$ is a weight matrix where $\mathrm{vol}(G) = \Sigma_u \, d_u$.

Let $\mathrm{u}_u \in \mathrm{R}^k$ e the embedding of node u and $U = (\mathrm{u1}, \ldots, \mathrm{u}n)^T \in \mathrm{R}^{n \times k}$ be the embedding matrix. POLE uses the dot product in the embedding space to preserve the signed autocovariance similarity R:

$$U^* = \arg\min_U \sum_{u,v} (\mathbf{u}_u^T \mathbf{u}_v - R_{uv})^2$$
$$= \arg\min_U \|UU^T - R\|_F^2$$

Which leads to a matrix factorization algorithm to find the optimal embedding. Specifically, $U^* = Q_k \sqrt{} * \mathrm{srqt}(\Lambda_k)$ —where $R = Q \Lambda Q^T$ is the Singular Value Decomposition (SVD) of R—is the optimal solution of U under the constraint $\mathrm{rank}(UU^T) = k$ [8].

SignedGCN [54] is a dedicated and principled effort that utilizes balance theory to correctly aggregate and propagate the information across layers of a signed GCN model.

To make it easier to understand how signedGCN works, we will show the differences with the conventional GCN described above.

When constructing a node representation in a traditional GCN, they aggregate their immediate local neighbors' information into a single representation and then propagate it around the network using multiple layers, allowing a node to incorporate information from a multi-hop neighborhood (where the number layers in the GCN denotes the number of hops away information is being aggregated from).

In signed networks, however, it cannot categorize all users in the same way. This is because semantically, users connected to a node via positive relationships are considered "friends," whereas neighbors connected via negative links are considered "enemy."

Instead of keeping a single representation for each node, the authors recommend that they keep a representation of both their "friends" and "enemies," which successfully integrates both the positive and negative links and provides a more comprehensive image of a certain user.

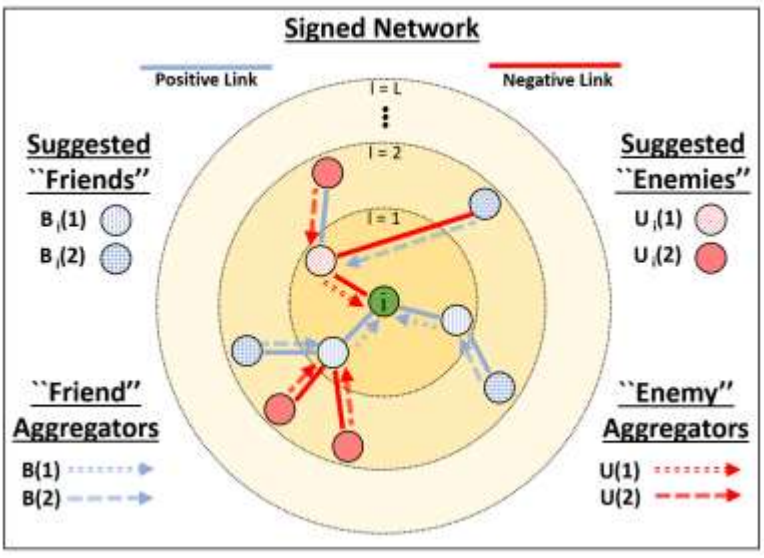


Figure 18: An illustration of how SGCN aggregates neighbor information in asigned network source[54]

Signed GCN maintains two representations at each layer, one for the corresponding balanced set of users (i.e., suggested "friends"), and one for the users in the respective unbalanced set (i.e., suggested "enemies")

The first aggregation layer (i.e, when l = 1), utilizes the following:

$$\mathbf{h}_i^{B(1)} = \sigma\left(\mathbf{W}^{B(1)}\left[\sum_{j\in\mathcal{N}_i^+}\frac{\mathbf{h}_j^{(0)}}{|\mathcal{N}_i^+|}, \mathbf{h}_i^{(0)}\right]\right)$$

$$\mathbf{h}_i^{U(1)} = \sigma\left(\mathbf{W}^{U(1)}\left[\sum_{k\in\mathcal{N}_i^-}\frac{\mathbf{h}_k^{(0)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{(0)}\right]\right)$$

here σ() is a non-linear activation function, $W^{B(1);}$ $W^{U(1)} \in R^{dout\times 2din}$ are the linear transformation matrices responsible for the "friends" and "enemies" coming from sets $B_i(1)$ and $U_i(1)$, respectively, and dout is the length of the two internal hidden representations. More specifically, for determining the hidden representation $h_i^{B(1)}$ it also concatenates the hidden representation of user ui (i.e., $h_i^{(0)}$) along with the mean of the users in set $B_i(1)$.

In all subsequent layers, the aggregation is more complex. The aggregations for l > 1 are defined as follows:

$$\mathbf{h}_i^{B(l)} = \sigma\left(\mathbf{W}^{B(l)}\left[\sum_{j\in\mathcal{N}_i^+}\frac{\mathbf{h}_j^{B(l-1)}}{|\mathcal{N}_i^+|}, \sum_{k\in\mathcal{N}_i^-}\frac{\mathbf{h}_k^{U(l-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{B(l-1)}\right]\right) \quad (5)$$

$$\mathbf{h}_i^{U(l)} = \sigma\left(\mathbf{W}^{U(l)}\left[\sum_{j\in\mathcal{N}_i^+}\frac{\mathbf{h}_j^{U(l-1)}}{|\mathcal{N}_i^+|}, \sum_{k\in\mathcal{N}_i^-}\frac{\mathbf{h}_k^{B(l-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{U(l-1)}\right]\right)$$

where $W^{B(l)}$; $W^{B(l)} \in R^{dout\times 3dout}$ for l > 1.

The SGCN objective function is the sum of two components, both of which help to understand the relationships between pairs of users in the signed network's embedded space.

The first term includes a weighted multinomial logistic regression (MLG) classifier as an additional layer. We want to know whether a pair of node embeddings are from users who have a positive, negative, or no link.

The second term is based on the extended structural balance theory. The purpose of the second term is for positively linked users to be closer in the embedded space than no link pairs, and no link pairs to be closer than users with a negative link between them. This term is controlled by λ to ensure that the contribution to the overall goal is balanced.

3.6. Embed Knowledge to Article Component.

Purpose: This component is responsible for the embedding of the polarization knowledge generated from the public graph into the polarization knowledge generated for each article.

After training the model and generating the embeddings for the nodes in the public graph, we embed this knowledge into the private graphs in the form of node embeddings.

We initially select which representations we are going to include. Next, for each of our private graphs we map the nodes from the public graph to the private graphs using the wikidata url and we then embed the selected embeddings to the appropriate node in the form of node embeddings.

Embeddings for Public Graph				
Donald_Trump	0.5	0.2	0.3	0.8
Joe_Biden	0.6	0.2	0.9	0.4
Kamala_Harris	0.2	0.4	0.3	0.6
Mike_Pence	0.8	0.1	0.4	0.8
World_Health_Organization	0.9	0.9	0.3	0.1

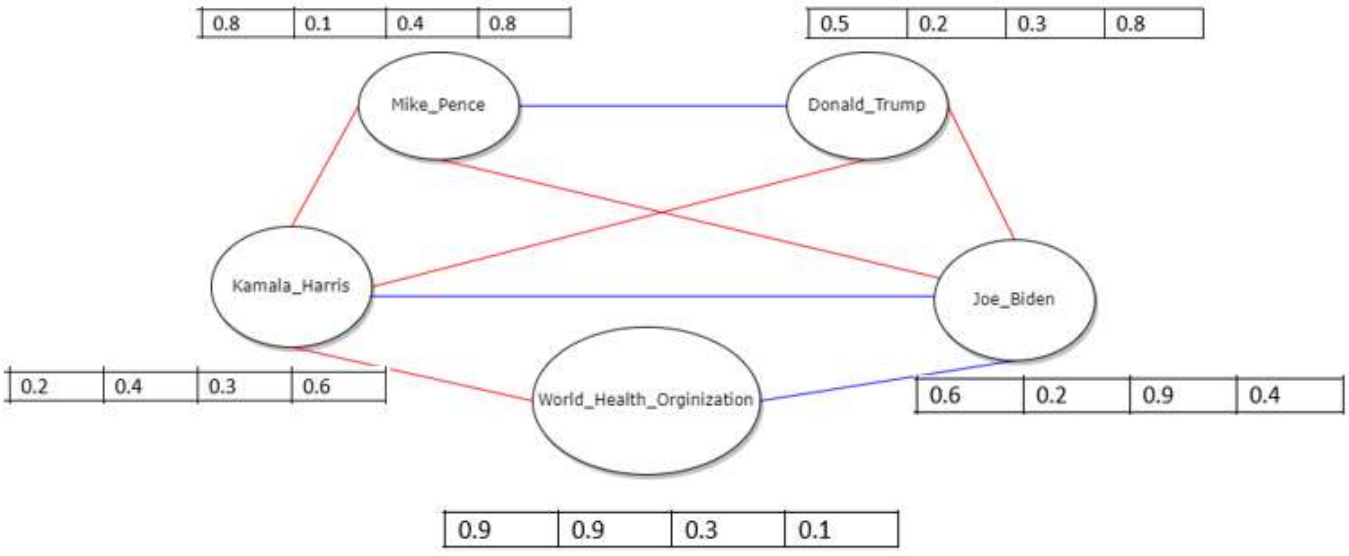


Figure 19: Example of how the Embed Knowledge to Article Component would work.

3.7. Private Graph Encoding Component

This component is responsible for transforming the knowledge of the private graphs, including the embedded knowledge from the public graph, into a single vector representation.

This component uses one of the encoding techniques indicted by the "Public Graph Encoding Component" but with a modification. Since we are now performing graph prediction, we need a way to represent the whole graph as a vector. In order to do that we use the TopKPooling pooling operation [76].

The topKpooling layer adaptively selects a subset of nodes to form a new but smaller graph. To this end, we employ a trainable projection vector p. By projecting all node features to 1D, we can perform k-max pooling for node selection. The k-max pooling operation outputs the k-largest units.

Since the selection is based on 1D footprint of each node, the connectivity in the new graph is consistent across nodes. Given a node I with its feature vector x_i , the scalar projection of xi on p is $y_i = xi*p/|p|$. Here, y_i measures how much information of node I can be retained when projected onto the direction of p. By sampling nodes, we wish to preserve as much information as possible from the original graph. To achieve this, we select nodes with the largest scalar projection values on p to form a new graph.

After each convolution operation, we use a topKpooling operation which leaves a subset of nodes, preserving as much information as possible from the original graph. Next, we get the global mean and max for each dimension of the embeddings and concatenate the two. Thus after each convolution-TopKpooling-global-pooling step we have a vector representation of the graph containing a subset of nodes of the previous step.

Finally, we aggregate the information of all steps by finding the mean of each dimension and this is the final vector representation for the graph.

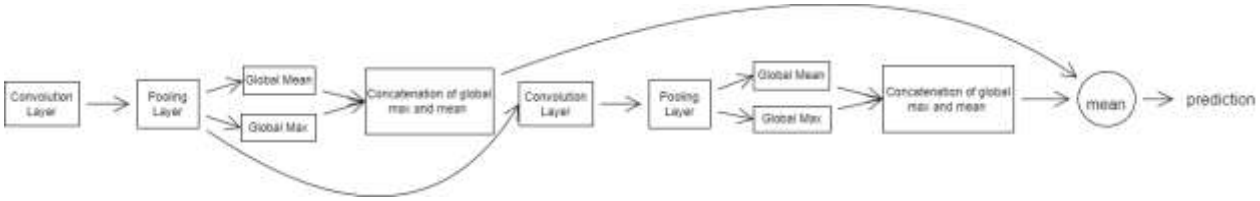


Figure 20: Visualization of two layered Private Graph Encoder

Chapter 4

Experiments & Evaluation

4.1. Dataset Discussion

For the purposes of our study, we have collected and labelled a new fake news detection dataset, whose theme is Covid19. This private dataset is collected from the GDELT Project[25] which is a large, comprehensive, and open database of global news articles and the subject of our study is COVID19.

We initially queried the GDELT Project for articles within the timeframe 21/1/2020 – 21-6-2021. We choose that particular time frame because that was the outburst of COVID19, and sadly colossal waves of misinformation followed. We then filtered these results to keep only the articles that were written in the United States.

After that, we found an open-source version of NewsGuard's [91] unreliable hosts, valid for our timeframe, and a list from all-sides[112] for the reliable hosts. Following we filtered the articles to only those that we had reliability information for their hosts, and labelled them accordingly.

Next that, we pre-processed the text from the remaining articles by first setting everything to lower case, removing stop words and digit tokens and lemmatizing the text.

Afterwards, we filtered the articles based on a number of COVID19 related keywords that we found to have appeared in the majority of COVID related news articles, giving the remaining articles the desired subject.

The result is a Covid19 Fake News Dataset with 161,933 articles, 143,070 of which is Reliable and 18,863 unreliable. This Dataset can be downloaded for free from out GitHub Repository.

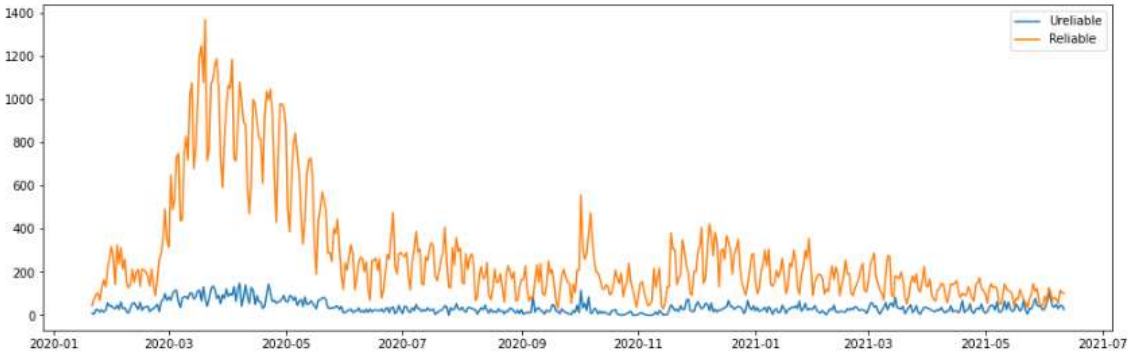


Figure 21: The figure above showcases how many reliable and unreliable articles are in our final private dataset as a function of time within our timeframe

4.2. Experimental Approaches Overview

In this section of the paper, we will discuss the base models that we used to examine the influence of our framework.

We have identified representative works in the literature and used these to examine how the addition of our work can aid each of them.

LSTM model (and other neural network approaches) [11]:

RNN-based approaches are by far the most common approach in fake news detection.

Recurrent Neural Network (RNN) is a feed-forward artificial neural network. RNNs handle a variable-length sequence input by comprising a recurrent hidden layer whose activation at each time is dependent on the previous time.

Long Short-Term Memory networks (LSTM) are a special type of RNN competent in learning long-term dependencies which are a very effective solution for addressing the vanishing gradient problem. Bi-Directional LSTM network steps through the input sequence in both directions at the same time.

We choose to use the recreate the work of Bahada et al in [11]. The Authors of this work have used a Bi-directional LSTM-recurrent neural network to make the classification.

The text is initially preprocessed by converting the text to UTF-8 format and by removing punctuation and stop words. Next, the news articles' titles and content text are turned into space-separated padded sequences of words, which are further split into lists of tokens.

Then the embedding layer will load the weights from the Global Vectors for Word Representation (GloVe) embeddings provided by the Stanford NLP team instead of loading random weights. GloVe applies globally aggregated co-occurrence statistics across all words in the news article corpus.

The resulting representations formalize significant linear substructures of the word vector space. This representation is then passed through the Bi-Directional LSTM which makes the classification

Sentiment analysis using RoBERTa [13]

A transformer is a deep learning model that uses the self-attention mechanism to weight the importance of each element of the input data differently. [77]

BERT is a language representation model that uses both left and right context conditioning in all layers to pretrain deep bidirectional representations from unlabelled text. It learns contextual relations between words (or sub-words) in a text using a multi-layer bidirectional Transformer encoder. BERT was created to be pre-trained once and then fine-tuned with just one additional output layer to produce state of the art models for a variety of tasks.[78]

RoBERTa is a robustly optimized version of BERT, that improves on the masked language modelling objective compared with BERT and leads to better downstream task performance. [79]

We find that this work by Kolevet al perfectly captures the spirit of style-based methods. Kolevet al initially trained a RoBERTa Model to perform Emotion Classification based on the six basic emotions based by the prominent psychologist Paul Ekman - fear, joy, anger, sadness, disgust, and surprise and a seventh emotional category was also added to represent neutral text.

The model was then used to make inferences on the emotion profile of news titles, returning a probability vector, which describes the emotion that the title conveys. Having the emotion probability vectors for each article's title, another Binary Random Forest classifier model was trained to evaluate news as either fake or real, based solely on their emotion profile.

We are going to make a slight modification and use an MLP classifier instead because all other modes use that classifier to make comparison easier.

Polarization Information Model

Finally, to further showcase how correlated polarization is with the task of fake news detection we have implemented and recorded the performance of a classifier using only the representation of Polarization information mentioned above.

Initially we generate the embeddings of the public graph using one of the methods described in section (3.5 Public Graph Encoding Component). Next, we embed the node embeddings for the specified representation into the private graphs and then use a Heterogeneous Graph Attention Network with the TopKPooling technique to generate the embeddings for the whole graph. Lastly, we use that representation to perform graph classification by forwarding it to an MLP to make the classification.

4.3. Results

We examined the influence of our work in each of the base models mentioned above by creating an ensemble model which concatenates the embeddings of the representation of the base model and the representation of the polarised information and then forwarded that through a classifier.

To keep things simple, we used a simple Multi-Layered Perceptron for the classifying part for every model, which makes the final classification reliable or unreliable. We made this choice because Neural Networks are by far the most popular option in the literature. It is easy to see that the user can substitute this with any other classifier of her choice if she so wishes.

The loss function used is the binary cross-entropy which is the de facto loss function for binary classification. The results of our experiments shown above were generated using 5-fold cross validation to get a more representative metric.

The experimental results for our models using Polarization knowledge can be summarised in table 1.

Model	Accuracy (%)	Precision	Recall	F1 - Score
Best GNN: GAT	0.621	0.588	0.717	0.646
Best KG: TranseR	0.615	0.584	0.736	0.648
Best Signed POLE	0.598	0.569	0.740	0.640
Best Combination Model without Signed. GAT + TransR	0.623	0.586	0.726	0.646
Best Combination Model GAT + TransR + POLE	0.642	0.663	0.631	0.647

Table 1: Experiment Results Using Only Polarization Knowledge

As it can be seen from table 1 all representation schemas where able to achieve competitive performance for the task of fake news detection. For reference the most common approach in the literature for fake news detection is a RNN based approach, and our implementation of it achieved 65% accuracy.

The best single encoder model was the Heterogenous GAT which was trained using the embeddings from the “Heterogeneous Graph with Edge Features” Shema. This was according to our expectations because it was our most expressive representation of the polarization knowledge.

The second-best accuracy for a single encoder model was from the TransR model using the “Knowledge Graph” Shema. It is worth mentioning that the average Knowledge Graph-based model outperformed its GNN counterparts in the more expressive “Heterogeneous Graph without Edge Features” Shema.

Moreover, it is clear that the difference results between this and the Heterogenous GAT are really small. These observations show that the Knowledge Graph Representation is particularly promising. Especially when taking into account the computational cost.

Lastly the POLE model using the “Sentiment Attribute Graph” Shema also managed to achieve compatible performance despite using only the raw polarization Information generated by our framework. The power of this representation however can be seen when looking at the combination models.

When we experimented with combinations of models using different representations, we found that the results for models that used schemas that included the majority of the generated information all had performance similar to the individual model. This hinted that the embeddings must capture similar information from the topology of the graph.

This was not true however for the combination models that used the “Sentiment Attribute Graph” schema and one of the more sophisticated schemas. The results of these models all averaged much higher that their individual models.

The best performance was achieved using a combination model of the “Heterogeneous Graph with Edge Features” , “Knowledge Graph” and the “Sentiment Attribute Graph” schemas, with the GAT, TransR and POLE model respectively. The final result is 0.64%, which is close to +2% performance, from each individual model. This is a noticeable increase for these models.

These findings clearly reveal a correlation between polarization and the detection of fake news, and they should motivate researchers into looking even further into other representations of polarization to help with the challenge of detecting fake news.

Model	Accuracy	Precision	Recall	F1 - Score
Bi-Directional LSTM	0.65	0.62	0.67	0.64
RoBERTa	0.83	0.85	0.83	0.84
Polarization	0.64	0.66	0.63	0.65
Polarization + Bi-Directional LSTM	0.76 (+0.11)	0.79 (+0.17)	0.72 (+0.05)	0.75 (+0.11)
Polarization + RoBERTa	0.87 (+0.04)	0.87 (+0.02)	0.88 (+0.05)	0.87 (+0.03)

Table 2: Experiment Results Summary

Table 2 summarizes the results of our experiments. The first conclusion we see from this table is how competitive our models using polarization knowledge are compared to the most common approach to fake news detection.

The Bi-directional LSTM only managed to outperform our model by 1% accuracy. The same conclusion can be seen from the rest of the metrics included as well. It is evident that the RoBERTa model clearly outperformed the rest of the single encoder models, by a respectable margin.

It is worth mentioning that despite the fact that the RoBERTa model clearly outperforms our Polarization based models, RoBERTa is considered to be a black box algorithm, even after applying state of the art explainable AI techniques. Our approach on the other hand clearly shows why that classification is made.

When looking at the combination models, which include a baseline encoding approach from the literature and our approach, it is clear that the improvement is compelling.

The improvement seen when combining our approach to the bi-directional LSTM is massive, with a +11% accuracy and an impressive +11% f1-score. This result clearly shows the immense potential of our approach.

The combination model that uses RoBERTa and our approach also shows an equally impressive feat. Despite the fact that the baseline RoBERTa model already has such a high performance, it was still able to further increase its performance to 87% accuracy a + 4% increase in accuracy.

Our findings suggest that all of the methods examined in this paper have a considerable performance boost in performance when using our framework. These results hint that our framework could benefit all existing fake news detection methods, including state of the art approaches like RoBERTa.

Conclusion & Future Work

5.1 Conclusion

In this paper we presented a novel approach to modelling polarization knowledge and displayed the first attempt to detect fake news using Polarization Knowledge.

Initially the presented framework creates a Knowledge Base of Public Polarization information about a subject within a specified time frame using one of our novel Graph representation schemas. It then uses this knowledge in combination with the polarization information of each article to classify it as reliable or unreliable.

We conducted extensive experiments with multiple representation of this information and multiple techniques for each, achieving a competitive performance. Next, we examined how this framework can be combined with representative state of the art approaches using different information and conducted extensive experiments showcasing its effectiveness.

Our findings show clear correlation between polarization and misinformation and open up a new category of fake news detection techniques.

We showed that POLAR can capture the necessary information for this, however there are multiple different representations of Polarization in the literature, each of which able to different aspects of polarization and thus aid in its own way at the task of fake news detection.

Moreover, our results show that all of the methods examined in this paper have a considerable performance boost, with the LSTM solution benefiting the most.

5.2 Future work

As future work we plan on applying this framework to the remaining types of fake news approaches, specifically knowledge based, propagation and Source-based methods.

Moreover, we plan on testing it to different datasets to further prove its robustness. Furthermore, this work uses POLAR to obtain the polarization information. As future work we plan on including different representations of polarization to see how they can be applied in this task.

Next, we plan on using Explainable AI techniques to further understand the relationship between the different representations of knowledge found in combination models using different Polarization knowledge schemas.

Another possible direction is to create different combinations of the 4+1 information sources generated from the different approaches of fake news detection and use Explainable AI techniques to examine how these combinations help each other.

Bibliography

1. C. colomina and H. S. Margalef , “The impact of disinformation on democratic processes and human rights in the world.” [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/STUD/2021/653635/EXPO_STU(2021)653635_EN.pdf. [Accessed: 24-May-2022].
2. Allcott, Hunt, and Matthew Gentzkow. "Social media and fake news in the 2016 election." Journal of economic perspectives 31.2 (2017): 211-36.
3. Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." ACM Computing Surveys (CSUR) 53.5 (2020): 1-40.
4. Brendan Nyhan and Jason Reifler. 2010. When corrections fail: The persistence of political misperceptions. Political Behavior 32, 2 (2010), 303–330.
5. Christopher Paul and Miriam Matthews. 2016. The Russian “Firehose of Falsehood” Propaganda Model. RAND Corporation (2016).
6. Rekker, Roderik. "The nature and origins of political polarization over science." *Public Understanding of Science* 30.4 (2021): 352-368.
7. Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." *ACM Computing Surveys (CSUR)* 53.5 (2020): 1-40.
8. Pan, Jeff Z., et al. "Content based fake news detection using knowledge graphs." *International semantic web conference*. Springer, Cham, 2018.
9. Bhutani, Bhavika, et al. "Fake news detection using sentiment analysis." *2019 twelfth international conference on contemporary computing (IC3)*. IEEE, 2019.
10. Calvert, Drew. “The Psychology behind Fake News.” *Kellogg Insight*, 20 Apr. 2022, https://insight.kellogg.northwestern.edu/article/the-psychology-behind-fake-news.
11. Bahad, Pritika, Preeti Saxena, and Raj Kamal. "Fake news detection using bi-directional LSTM-recurrent neural network." *Procedia Computer Science* 165 (2019): 74-82.
12. Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.
13. Kolev, Vladislav, Gerhard Weiss, and Gerasimos Spanakis. "FOREAL: RoBERTa Model for Fake News Detection based on Emotions." *The 14th International Conference on Agents and Artificial Intelligence*. Scitepress-Science And Technology Publications, 2022.
14. Pan, Jeff Z., et al. "Content based fake news detection using knowledge graphs." *International semantic web conference*. Springer, Cham, 2018.
15. Guha, Ramanthan, et al. "Propagation of trust and distrust." *Proceedings of the 13th international conference on World Wide Web*. 2004.
16. Heider, Fritz. "Attitudes and cognitive organization." *The Journal of psychology* 21.1 (1946): 107-112.

17. Han, Yi, et al. "Knowledge Enhanced Multi-modal Fake News Detection." *arXiv preprint arXiv:2108.04418* (2021).
18. Ren, Yuxiang, et al. "Adversarial active learning based heterogeneous graph neural network for fake news detection." *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020.
19. Paschalides, Demetris, et al. "Check-It: A plugin for detecting and reducing the spread of fake news and misinformation on the web." *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2019.
20. Mayank, Mohit, Shakshi Sharma, and Rajesh Sharma. "DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection." *arXiv preprint arXiv:2107.10648* (2021).
21. Guerra, Pedro, et al. "A measure of polarization on social media networks based on community boundaries." *Proceedings of the international AAAI conference on web and social media*. Vol. 7. No. 1. 2013.
22. Conover, Michael, et al. "Political polarization on twitter." *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 5. No. 1. 2011.
23. Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." *ACM Computing Surveys (CSUR)* 53.5 (2020): 1-40.
24. Paschalides, Demetris, George Pallis, and Marios D. Dikaiakos. "POLAR: a holistic framework for the modelling of polarization and identification of polarizing topics in news media." Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2021.
25. "The GDELT Project." *GDELT*, https://www.gdeltproject.org/.
26. Sitaula, Niraj, et al. "Credibility-based fake news detection." *Disinformation, Misinformation, and Fake News in Social Media*. Springer, Cham, 2020. 163-182.
27. Del Vicario, Michela, et al. "The spreading of misinformation online." *Proceedings of the National Academy of Sciences* 113.3 (2016): 554-559.
28. Garimella, Kiran, et al. "Quantifying controversy on social media." *ACM Transactions on Social Computing* 1.1 (2018): 1-27.
29. Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In Proc. of LIkKDD, page 36–43, NY, USA, 2005. ACM
30. Aral, Sinan, and Dean Eckles. "Protecting elections from social media manipulation." *Science* 365.6456 (2019): 858-861.
31. Zollo, Fabiana, et al. "Debunking in a world of tribes." *PloS one* 12.7 (2017): e0181821.
32. "NLP - Expand Contractions in Text Processing." *GeeksforGeeks*, 21 Feb. 2022, https://www.geeksforgeeks.org/nlp-expand-contractions-in-text-processing/.
33. Ganesan, Kavita. "What Are Stop Words?" *Kavita Ganesan, PhD*, 30 July 2020, https://kavita-ganesan.com/what-are-stop-words/#.YpDUWChBy3A.
34. *Stemming and Lemmatization*, https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

35. Zheng, Jiaping, et al. "Coreference resolution: A review of general methodologies and applications in the clinical domain." *Journal of biomedical informatics* 44.6 (2011): 1113-1122.
36. “Main Page.” *Wikidata*, https://www.wikidata.org/wiki/Wikidata:Main_Page.
37. Parravicini, Alberto, et al. "Fast and accurate entity linking via graph embedding." *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. 2019.
38. Perozzi, Bryan, et al. "Online learning of social representations." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi. org/10.1145/2623330.2623732*.
39. Choi, Eunsol, et al. "Document-level sentiment inference with social, faction, and discourse context." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016.
40. Toledo-Ronen, Orith, et al. "Learning sentiment composition from sentiment lexicons." *Proceedings of the 27th International Conference on Computational Linguistics*. 2018.
41. Tang, Jiliang, et al. "A survey of signed network mining in social media." *ACM Computing Surveys (CSUR)* 49.3 (2016): 1-37.
42. Esmailian, Pouya, and Mahdi Jalili. "Community detection in signed networks: the role of negative ties in different scales." *Scientific reports* 5.1 (2015): 1-17.
43. Tang, Jiliang, et al. "A survey of signed network mining in social media." *ACM Computing Surveys (CSUR)* 49.3 (2016): 1-37.
44. Cartwright, Dorwin, and Frank Harary. "Structural balance: a generalization of Heider's theory." *Psychological review* 63.5 (1956): 277.
45. Aref, Samin, and Zachary Neal. "Detecting coalitions by optimally partitioning signed networks of political collaboration." *Scientific reports* 10.1 (2020): 1-10.
46. Aref, Samin, and Mark C. Wilson. "Balance and frustration in signed networks." *Journal of Complex Networks* 7.2 (2019): 163-189.
47. Aref, Samin, and Mark C. Wilson. "Balance and frustration in signed networks." *Journal of Complex Networks* 7.2 (2019): 163-189.
48. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems* 26 (2013).
49. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
50. Von Luxburg, Ulrike. "A tutorial on spectral clustering." *Statistics and computing* 17.4 (2007): 395-416.
51. Wang, Xiao, et al. "Heterogeneous graph attention network." *The world wide web conference*. 2019.
52. Web.stanford.edu. 2022. What is a Knowledge Graph?. [online] Available at: <https://web.stanford.edu/~vinayc/kg/notes/What_is_a_Knowledge_Graph.html> [Accessed 27 May 2022].

53. Glasmachers, Tobias. "Limits of end-to-end learning." *Asian Conference on Machine Learning*. PMLR, 2017.
54. Derr, Tyler, Yao Ma, and Jiliang Tang. "Signed graph convolutional networks." *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018.
55. Leskovec, Jure, Daniel Huttenlocher, and Jon Kleinberg. "Signed networks in social media." *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010.
56. Ji, Shaoxiong, et al. "A survey on knowledge graphs: Representation, acquisition, and applications." *IEEE Transactions on Neural Networks and Learning Systems* (2021).
57. Bordes, Antoine, et al. "Translating embeddings for modeling multi-relational data." *Advances in neural information processing systems* 26 (2013).
58. Lin, Yankai, et al. "Learning entity and relation embeddings for knowledge graph completion." *Twenty-ninth AAAI conference on artificial intelligence*. 2015.
59. He, Shizhu, et al. "Learning to represent knowledge graphs with gaussian embedding." *Proceedings of the 24th ACM international on conference on information and knowledge management*. 2015.
60. S. Kullback, Information Theory and Statistics. North Chelmsford, MA, USA: Courier Corporation, 1997
61. Jenatton, Rodolphe, et al. "A latent factor model for highly multi-relational data." *Advances in neural information processing systems* 25 (2012).
62. Yang, Bishan, et al. "Embedding entities and relations for learning and inference in knowledge bases." *arXiv preprint arXiv:1412.6575* (2014).
63. Trouillon, Théo, et al. "Complex embeddings for simple link prediction." *International conference on machine learning*. PMLR, 2016.
64. Wang, Quan, et al. "Knowledge graph embedding: A survey of approaches and applications." *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017): 2724-2743.
65. Wang, Zhen, et al. "Knowledge graph embedding by translating on hyperplanes." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. No. 1. 2014.
66. Bouchard, Guillaume, Sameer Singh, and Theo Trouillon. "On approximate reasoning capabilities of low-rank vector spaces." *2015 AAAI Spring Symposium Series*. 2015.
67. Hamilton, William L., Rex Ying, and Jure Leskovec. "Representation learning on graphs: Methods and applications." *arXiv preprint arXiv:1709.05584* (2017).
68. Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.
69. Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).
70. Veličković, Petar, et al. "Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017).

71. “Heterogeneous Graph Learningℑ.” *Heterogeneous Graph Learning - pytorch_geometric Documentation*, https://pytorch-geometric.readthedocs.io/en/latest/notes/heterogeneous.html.
72. Sergios Karagiannakos. “Best Graph Neural Network Architectures: GCN, Gat, MPNN and More.” *AI Summer*, Sergios Karagiannakos, 23 Sept. 2021, https://theaisummer.com/gnn-architectures/.
73. Cristina, Stefania. “The Attention Mechanism from Scratch.” *Machine Learning Mastery*, 20 Sept. 2021, https://machinelearningmastery.com/the-attention-mechanism-from-scratch/.
74. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).
75. Huang, Zexi, Arlei Silva, and Ambuj Singh. "POLE: Polarized Embedding for Signed Networks." *arXiv preprint arXiv:2110.09899* (2021).
76. Gao, Hongyang, and Shuiwang Ji. "Graph u-nets." *international conference on machine learning*. PMLR, 2019.
77. “Transformer (Machine Learning Model).” *Wikipedia*, Wikimedia Foundation, 25 May 2022, https://en.wikipedia.org/wiki/Transformer_(machine_learning_model).
78. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
79. Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
80. (2022). Retrieved 27 May 2022, from https://www.reuters.com/
81. (2022). Retrieved 27 May 2022, from https://www.wsj.com/
82. Breaking News, World News and Video from Al Jazeera. (2022). Retrieved 27 May 2022, from https://www.aljazeera.com/
83. ByRoland Oliphant, i., Andrews, K., Lilico, A., Mordaunt, P., Woods, J., & Nelson, F. et al. (2022). Telegraph. Retrieved 27 May 2022, from https://www.telegraph.co.uk/
84. CNN International - Breaking News, US News, World News and Video. (2022). Retrieved 27 May 2022, from https://edition.cnn.com
85. Home - BBC News. (2022). Retrieved 27 May 2022, from https://www.bbc.co.uk/news
86. https://www.washingtontimes.com, T. (2022). Washington Times - Politics, Breaking News, US and World News. Retrieved 27 May 2022, from https://www.washingtontimes.com/
87. NBC News - Breaking News & Top Stories - Latest World, US & Local News. (2022). Retrieved 27 May 2022, from https://www.nbcnews.com/
88. News, A. (2022). ABC News – Breaking News, Latest News, Headlines & Videos. Retrieved 27 May 2022, from https://abcnews.go.com/
89. The New York Times - Breaking News, US News, World News and Videos. (2022). Retrieved 27 May 2022, from https://www.nytimes.com/https://www.washingtonpost.com/

90. NewsGuard - Combating Misinformation with Trust Ratings for News. NewsGuard. https://www.newsguardtech.com/. Published 2022. Accessed May 27, 2022.
91. Khyani, Divya, et al. "An Interpretation of Lemmatization and Stemming in Natural Language Processing." *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology* 22 (2020): 350-357.
92. "Aiimi Labs On… Named-Entity Recognition". Aiimi, 2022, https://www.aiimi.com/insights/aiimi-labs-on-named-entity-recognition.
93. "Entity Linking - Wikipedia". En.Wikipedia.Org, 2022, https://en.wikipedia.org/wiki/Entity_linking. Wang, Shuai, et al. "Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts." *IEEE Access* 7 (2019): 136951-136961.
94. Gao, Hongyang, and Shuiwang Ji. "Graph u-nets." *international conference on machine learning*. PMLR, 2019.
95. References
96. 2022, https://www.merriam-webster.com/words-at-play/the-real-story-of-fake-news.
97. "A Brief History Of Fake News | Center For Information Technology And Society - UC Santa Barbara". Cits.Ucsb.Edu, 2022, https://www.cits.ucsb.edu/fake-news/brief-history.
98. Beaujon, Andrew. "Trump Claims He Invented The Term "Fake News"—Here' S An Interview With The Guy Who Actually Helped Popularize It - Washingtonian". Washingtonian - The Website That Washington Lives By., 2022, https://www.washingtonian.com/2019/10/02/trump-claims-he-invented-the-term-fake-news-an-interview-with-the-guy-who-actually-helped-popularize-it/.
99. SOLL, JACOB et al. "The Long And Brutal History Of Fake News". POLITICO Magazine, 2022, https://www.politico.com/magazine/story/2016/12/fake-news-history-long-violent-214535/.
100. Bovet, Alexandre, and Hernán A. Makse. "Influence of fake news in Twitter during the 2016 US presidential election." *Nature communications* 10.1 (2019): 1-14.
101. "'Post-Truth' Declared Word Of The Year By Oxford Dictionaries". BBC News, 2022, https://www.bbc.com/news/uk-37995600.
102. Choi, Yoonjung, Yuchul Jung, and Sung-Hyon Myaeng. "Identifying controversial issues and their sub-topics in news articles." *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer, Berlin, Heidelberg, 2010.
103. Esuli, Andrea, and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." *Proceedings of the fifth international conference on language resources and evaluation (LREC'06)*. 2006.
104. Mejova, Yelena, et al. "Controversy and sentiment in online news." *arXiv preprint arXiv:1409.8152* (2014).
105. Gruzd, Anatoliy, and Jeffrey Roy. "Investigating political polarization on Twitter: A Canadian perspective." *Policy & internet* 6.1 (2014): 28-45.
106. Theaisummer.Com, 2022, https://theaisummer.com/gnn-architectures.

107. "Aiimi Labs On… Named-Entity Recognition". Aiimi, 2022, https://www.aiimi.com/insights/aiimi-labs-on-named-entity-recognition.
108. "Two Minutes NLP—Quick Intro To Coreference Resolution With Neuralcoref". Medium, 2022, https://medium.com/nlplanet/two-minutes-nlp-quick-intro-to-coreference-resolution-with-neuralcoref-7fa2be2c4284.
109. Wooldridge, Mike, and Mike Wooldridge. "Making New Connections With Marklogic Semantics - Marklogic". Marklogic, 2022, https://www.marklogic.com/blog/making-new-connections-ml-semantics/.
110. Yang, Xu, et al. "Research of personalized recommendation technology based on knowledge graphs." *Applied Sciences* 11.15 (2021): 7104.
111. "Allsides | Balanced News Via Media Bias Ratings For An Unbiased News Perspective". Allsides, 2022, https://www.allsides.com/unbiased-balanced-news.

Appendix

Reliable hosts:

Reuters [80],	Aljazeera[84]	BBC News[88]
The New York Times [81]	ABC News[85]	The Wall Street Journal [89]
NBC News [82]	Washington Times[86]	The Washington Post[90]
The Telegraph[83]	CNN[87]	New York Post[91]

Unreliable hosts:

breitbart	activistpost.com	americanthinker.com	beforeitsnews.com
bigleaguepolitics.com	dcclothesline.com	greatgameindia.com	greenmedinfo.com
healthimpactnews.com	healthnutnews.com	humansarefree.com	infowars.com
intellihub.com	jimbakkershow.com	jimhumble.co	mercola.com
naturalhealth365.com	redstatewatcher.com	rushlimbaugh.com	sott.net
thebl.com	theepochtimes.com	themindunleashed.com	thetruthaboutcancer.com
wakingtimes.com	wnd.com	zerohedge.com	naturalnews.com network
naturalnews.com	banned.news	biased.news	californiacollapse.news
cdc.news	censorship.news	conspiracy.news	cures.news
depopulation.news	disinfo.news	eugenics.news	extinction.news
factcheck.news	faked.news	freedom.news	health.news
herbs.news	honest.news	infections.news	journalism.news
mediafactwatch.com	medicalextremism.com	medicine.news	naturalcures.news

naturalnewsradio.com	naturopathy.news	newsfakes.com	newstarget.com
nytwatch.com	openborders.news	outbreak.news	pandemic.news
panic.news	plantmedicine.news	populationcontrol.news	propaganda.news
realinvestigations.news	remedies.news	risk.news	scienceclowns.com
sciencefraud.news	science.news	scientific.news	shtf.news
superbugs.news	techgiants.news	technocrats.news	twisted.news
tyranny.news	uprising.news	vaccinedamage.news	vaccineinjurynews.com
vaccines.news	wapoop.news	washingtonposted.news	presstv.com
aubedigitale.com	epochtimes.fr	fl24.net	fr.sputniknews.com
french.presstv.com	lemediapourtous.fr	lesmoutonsenrages.fr	lesmoutonsrebelles.com
lumieresurgaia.com	nosignalfound.fr	nouvelordremondial.cc	patriote.info
reseauinternational.net	ripostelaique.com	wikistrike.com	affaritaliani.it
caffeinamagazine.it	corvelva.it	disinformazione.it	ilpopulista.it
ilprimatonazionale.it	it.sputniknews.com	leggilo.org	maurizioblondet.it
mednat.org	renovatio21.com	scenarieconomici.it	segnidalcielo.it
stopcensura.info	tgcom24.mediaset.it	voxnews.info	anonymousnews.ru
compact-online.de	connectiv.events	de.sputniknews.com	deutsch.rt.com

deutschland-kurier.org	indexexpurgatorius.wordpress.com	journalistenwatch.com	news-for-friends.de
politikversagen.net	pravda-tv.com	watergate.tv	

COVID19 related keywords

covid	virus	antibod	remdesevir	gates	lockdown
corona	zeneca	vaccine	hydroxychlo	immun	wuhan
coronavirus	moderna	vax	infect	mask	quarantin
pandemic	pfizer	sars			