National Technical University of Athens

School of Electrical
and Computer Engineering

Computer Science Division

Leveraging social networks and knowledge graphs for discovering and recommending engaging and credible information

Thesis

submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

by

Danae Pla Karidi

Diploma in Electrical and Computer Engineering
National Technical University of Athens

Athens, January 2022

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολη Ηλεκτρολογων Μηχανικων
Και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αξιοποιηση κοινωνικων δικτυων και γραφων γνωσης για την ευρεση και συσταση ενδιαφερουσας και αξιοπιστης πληροφοριας

Διδακτορική Διατριβή

της

Δανάης Πλα Καρύδη

Διπλωματούχου Ηλεκτρολόγου Μηχανικού & Μηχανικού Υπολογιστών
Εθνικού Μετσοβίου Πολυτεχνείου

Αθήνα, Ιανουάριος 2022

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων Και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αξιοποίηση κοινωνικών δικτύων και γράφων γνώσης για την εύρεση και σύσταση ενδιαφέρουσας και αξιόπιστης πληροφορίας

Διδακτορική Διατριβή

της

Δανάης Πλα Καρύδη

Διπλωματούχου Ηλεκτρολόγου Μηχανικού & Μηχανικού Υπολογιστών
Εθνικού Μετσοβίου Πολυτεχνείου

Συμβουλευτική Επιτροπή: Ι. Βασιλείου
Θ. Δαλαμάγκας
Α. Γ. Σταφυλοπάτης

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 26η Ιανουαρίου 2022.

Ι.Βασιλείου Ομότ. Καθ. ΕΜΠ	Α. Γ. Σταφυλοπάτης Καθ. ΕΜΠ	Θ. Δαλαμάγκας Ερευνητής Α΄ Ε. Κ. ΑΘΗΝΑ
Ε. Χριστοφόρου Καθ. ΕΜΠ	Δ.Τσουμάκος Καθ. ΕΜΠ	Ε.Συκάς Καθ. ΕΜΠ
Ι. Σταύρακας Ερευνητής Α΄ Ε. Κ. ΑΘΗΝΑ		

Αθήνα, Ιανουάριος 2022

...

Δανάη Πλα Καρύδη
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

...

Danae Pla Karidi
Doctor of Philosophy at the National Technical University of Athens

ΠΕΡΙΛΗΨΗ

Η αυξανόμενη χρήση των μέσων κοινωνικής δικτύωσης αναδεικνύει τον κρίσιμο ρόλο των αλγορίθμων κατάταξης και προτάσεων περιεχομένου που επηρεάζουν την εμπειρία των χρηστών και την ενασχόλησή τους με ενδιαφέρον περιεχόμενο. Επιπλέον, η αυξανόμενη δραστηριότητα στα μέσα κοινωνικής δικτύωσης επιταχύνει και ενισχύει την παραπληροφόρηση και τη διάδοση ψευδών ειδήσεων, επισημαίνοντας την ανάγκη για την ανάπτυξη αποτελεσματικών μοντέλων ανίχνευσης παραπλανητικού περιεχομένου. Εκτός από τα κοινωνικά δίκτυα, η ψηφιακή μετάβαση δημιουργεί σημαντική αύξηση και ποικιλομορφία των διαθέσιμων συνόλων δεδομένων τα οποία περιλαμβάνουν πλούσιο και ετερογενές περιεχόμενο. Ως εκ τούτου, ο όγκος και η πολυπλοκότητα των διαθέσιμων δεδομένων καθιστούν τα αποδοτικά συστήματα εξερεύνησης δεδομένων ολοένα και πιο κρίσιμα. Παράλληλα, οι μηχανικοί γνώσης οργανώνουν τα αντικείμενα της ανθρώπινης γνώσης και τις σχέσεις μεταξύ τους για τη δημιουργία γράφων γνώσης διαχειρίσιμων από υπολογιστικά συστήματα.

Σε αυτή τη διατριβή, διερευνήσαμε τις δυνατότητες σύνδεσης μεταξύ αυτών των ερευνητικών περιοχών. Ανακαλύψαμε ότι ο συνδυασμός των γνώσεων που παρέχονται από τους πιο προηγμένους γράφους γνώσης και δεδομένων από κοινωνικά δίκτυα μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα των συστημάτων προτάσεων. Επιπλέον, οι σχέσεις που εξάγονται από βάσεις γνώσεων όπως οι υπηρεσίες ελέγχου γεγονότων μπορούν να παρέχουν σύνολα δεδομένων εκπαίδευσης για την ανάπτυξη αποτελεσματικών μοντέλων ανίχνευσης παραπληροφόρησης, ενώ τα σημασιολογικά στοιχεία των γράφων γνώσης μπορούν να βοηθήσουν στην αποτελεσματική εξερεύνηση συνόλων δεδομένων και η ανάλυση δεδομένων από τα κοινωνικά δίκτυα μπορεί να συμβάλλει στην περιήγηση και την κατανόηση των διαθέσιμων δεδομένων.

Λέξεις-κλειδιά: Ανάλυση Κοινωνικών Δικτύων, Συστήματα Προτάσεων βάσει Περιεχομένου, Γράφοι Γνώσης, Ψευδείς Ειδήσεις, Αυτόματη Δημιουργία Συνόλων Δεδομένων, Ιδιότητες Διάδοσης, Σημασιολογικές Ιδιότητες, Εναλλακτικές Μετρικές Απήχησης, Ανάκτηση Πληροφοριών, Εξερεύνηση δεδομένων, Γεωγραφική Απεικόνιση

ABSTRACT

The increasing use of social media brings up the crucial role of social content recommendation and ranking algorithms that affect user engagement to interesting content. Additionally, the growing social network activity accelerates and boosts misinformation and disinformation diffusion, highlighting the need for efficient fake news detection models. Apart from social networks, digital transformation results in significant data growth and diversity of available datasets that result in rich and heterogeneous content. Hence, the volume and complexity of available data make efficient data exploration systems more and more crucial. On a separate front, knowledge engineers organize the objects of human knowledge and the relationships between them to create computer-manageable knowledge graphs.

In this thesis, we explored the possibilities of connections between these research areas. We showed that combining the knowledge provided by the most advanced knowledge graphs and data from social networks can significantly improve the efficiency of recommendation systems. Furthermore, relations extracted from knowledge bases such as fact-checking services can provide ground truth training datasets to develop efficient misinformation detection models, while the semantic elements of knowledge graphs can assist in exploring datasets effectively, and the analysis of data from social networks can contribute to browsing and understanding the available data.

Keywords: Social Network Analysis, Content-based Recommender Systems, Knowledge Graphs, Fake News, Automatic Dataset Generation, Propagation Features, Semantic Features, Altmetrics, Information Retrieval, Data Exploration, Geographic visualization

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The main interest of our research has been in recommending engaging and discovering credible information spreading in social networks. To this end, we focused on understanding the role of semantic content, graph properties, and patterns of information diffusion in social networks.

How can we recommend interesting content to social network users? How can we spot early the diffusion of fake news in a time-evolving network? How can we overcome natural language ambiguity to explore large datasets efficiently? How can we leverage social networks to discover credible information?

Answers to such questions are vital to a range of application areas, from developing new algorithms for recommending engaging content on social networks and other platforms, monitoring fake-news diffusion, developing knowledge-based data exploration techniques, to discovering and ranking emerging literature articles based on social media attention.

A basic intuition behind our study was using knowledge graphs as a solid knowledge background to exploit semantic relations between users' interests and data objects. Towards the implementation of the semantic web, knowledge engineers organize human knowledge in a formal and, at the same time, essential way. In this context, knowledge graphs are a vital structure for organizing the objects of human knowledge and the relationships between them, with semantically structured information so that computer systems can manage it. It is no coincidence that the most influential corporate groups in IT promote the development of such structures to improve their commercial services (navigation, search, personalized offers, targeted advertising) and increase user engagement. The nodes of knowledge graphs represent specific objects of knowledge, and the edges represent the relationships between them; that is, they focus on representing specific events, people, entities, objects, regions, and relationships between them.

In this thesis, we explored the possibilities of connections between these research areas:

- **Recommending engaging content in social networks**

  The increasing use of social media has brought about changes in the way a large part of humanity communicates, stays informed, works, shapes its perception of the world and social phenomena. For example, Twitter users generate hundreds of millions of tweets every day, leading to a vast amount of available information that ends up in other users' timelines. Because of this content overload, users are often tired of browsing for engaging tweets

on their timelines, and as a result, they miss interesting tweets. A solution to this problem can be developing efficient recommender systems that help users filter out uninteresting tweets and suggest ranked tweets and users with relevant interests, resulting in a more engaging timeline.

When analyzing the propagation of social network content, scientists face a severe problem: the algorithm that ranks the posts in the users' timeline is usually unknown; however, its role in the information diffusion process is crucial. Moreover, the streaming information regarding user activity is a commodity used to effectively expose a target group to a specific commercial message. Thus, social media platforms model users' behavior, predict and learn from it, take advantage of people's preferences, interests, and habits, offer them interesting content, and thus increase engagement on the platform and commercial content.

In addition, the recommendation algorithms of social networking services are not static; instead, they become more and more complex based on the research and development of new recommendation models and user profiling techniques. However, their closeness makes it unclear whether they serve to provide "what each user is interested in seeing" or whether they are more oriented to prioritize "what each user should see" to meet various requirements. Moreover, it is impossible to evaluate the personalization process they follow and whether they facilitate the expansion of misleading phenomena (fake news, bots, trolls, etc.). Therefore, the evolution of recommendation algorithms is of great interest, and one such typical example is Twitter's algorithm. Twitter's algorithm, like most social media algorithms, is based on personalization and machine learning to rank content by some signals: recency, relevance, engagement, rich media, and the unfolding past user behavior (likes, clicks, retweets, popularity, region, etc.). Even though such algorithms canalize the attention of millions of users, we still know little about them.

In this context, recommendation algorithms may limit the user perspective by recommending content similar to what they usually post or engage with. Hence, users are exposed to certain information, and they interact with news that likely promotes their favored opinions, resulting in the formation of social groups with like-minded people (echo chamber communities). The *echo chamber* effect influences how the news is consumed and enhances the propagation of misleading content, acting as fake news reinforcement hubs.

- **Discovering credible information in social networks** The widespread use of social media as a news and information source has posed some new challenges. Although misinformation and disinformation phenomena have existed since the birth of the printed press, new online platforms accelerate and boost their diffusion, posing new problems and challenges. However, contrary to traditional news organizations, social media platforms allow millions of users to produce and access a vast source of information freely. On the ground of much faster news diffusion, the relative anonymity offered by social media, and the overt and covert mechanisms that operate on the web serving political, economic, geopolitical, and other interests, the phenomenon of fake news diffusion has evolved and acquired particular features related to the propagation in social media.

The fight against fake news is a subject that social network engineers should focus on. However, the sources and causes of misinformation and disinformation prevail in the spectrum of social phenomena. This explains why many rumors survive for a long time after being exposed, why, despite the rise of the educational level of humanity, unscientific theories and prejudices survive. The reason is that the human learns in a social context, adopts and constructs a large part of her perceptions based on the data given to her as undisputed knowledge by people and sources she trusts (teachers, parents, friends, sources of information, etc.). In some cases, the misinformation starts from specific news pages, and through the propagation boosting of social media, it even finds a place in the pages of generally considered credible newspaper websites.

Moreover, state intelligence services and most armies have special directorates and divisions that engage in cyber warfare, a central aspect of which is propaganda and the spreading of misinformation to confuse and deceive the enemy. Thus, sometimes behind fake news lies a "war" of state and business interests, a "war" that serves not only the purposes of propaganda but also has serious economic and social consequences. In any case, there are many recent publications [172, 131, 25], according to which social networking data are gathered in the hands of governments and private companies, forming the requirements toward the existence of mass surveillance systems. Based on the above, it is expected for social media to be used as political tools to form the so-called "public opinion", manipulating consciences, in essence, to increase the influence of states, political parties, business groups. In this context, the opposing "armies" often call the activity of another "misinformation" or "hate speech", resulting in many cases concerning censorship allegations in the name of "fighting misinformation and hate speech". Besides, many fact-checking organizations and services have been accused of bias, while the effect of fact-checking can vary by several factors [99].

Nevertheless, the development of methods and algorithms that can detect suspicious and misleading content and produce early detection signals offers a significant contribution to limiting and fighting the further spread of such content. However, no system can cure an infodemic, as the sources and causes of the misinformation phenomenon remain social rather than technical.

Apart from the vast amount of social network data, big data results in significant data growth and, despite the lower availability, increases the need for efficient data exploration systems, under the light of which new possibilities are brought out in the development of production, the organization of work, the progress of science. The size and diversity of available datasets, their continuous collection and generation by systems, sensors, scientists, organizations, government, and online content growth result in rich and heterogeneous content. Hence, the volume and complexity of available data make efficient data exploration systems more and more crucial.

In this thesis, we studied the connections of these research areas, set on the solid semantic background of knowledge graphs. For example, combining the knowledge provided by the most advanced knowledge graphs and data from social networks can significantly improve the efficiency of recommendation systems. Furthermore, relations extracted from knowledge bases such as fact-checking services can provide

ground truth training datasets to develop efficient misinformation detection models, and in turn, these detection models can help cleaning existing knowledge graphs from mistakes and false information. In addition, the semantic elements of knowledge graphs can assist in exploring datasets effectively, and the analysis of data from social networks can contribute to browsing and understanding the available data.

1.1 Thesis overview

1.1.1 Tweet and followee personalized recommendations based on knowledge graphs

Recommending content and connections on social networks faces multiple challenges. First, user profiling is often undermined by the limitations posed by social networking services concerning the availability of user network data. Therefore, techniques that construct profiles based on user social connections (collaborative filtering) require a large volume of link data to be retrieved, stored, and analyzed and thus cannot be updated effectively and scalable when new tweets enter the stream. Moreover, collaborative filtering approaches will most likely evaluate similar items unequally if posted by different users, despite having the same content. These approaches also require each item to get instant feedback from numerous users before being recommended to other users, known as the "cold-start" problem. A different approach, the content-based recommenders, relies on text similarity; however, it lacks efficiency in microblogging services due to small text size.

Our view is that the semantic features of knowledge graphs can be helpful in user profiling and analyzing social network data. However, in the case of Twitter, the content of tweets is small and sparse; hence a topic analysis using bag-of-words approaches remains insufficient to reflect user interests accurately. Moreover, content-based techniques must balance between over-recommendation, i.e., recommending too many items, and over-specialization, i.e., recommending items very similar to those already seen, containing recurrent information, and not covering one's range of interests.

Our study focused on developing a new content-based method [72] that uses knowledge graphs for (a) personalized tweet recommendations [71] and (b) personalized followee recommendations [69]. This method provides an alternative personalized timeline containing worldwide streaming tweets that strongly match the user's interests and personalized recommendations of followees ("whom to follow?") with similar interests.

Our approach utilizes the semantic relevance between user interests and the topics of streaming tweets. Choosing semantic relevance as the recommendation criterion has the following advantages. First, considering that users read and write content over multiple topics, our recommendation method utilizes the objective and immutable associations between these topics. Moreover, our method tackles the over-specialization problem by taking advantage of the knowledge graph to recommend tweets of related topics and the over-recommendation problem by ranking the user profiles, emphasizing the most specific topics. In contrast to collaborative filtering techniques, our approach does not face the problem of resource availability because it makes no use of the Twitter user graph data. Finally, contrary to the bag-of-words methods (LDA, TF-IDF), our approach avoids the efficiency problems

caused by the limited size of tweets. Moreover, we assign a topic to each tweet separately in contrast to most content-based methods, which merge tweets and defy the proper granularity for topic extraction.

Our approach is based on representing all possible user interests as a hierarchical knowledge graph, where each node corresponds to a topic, and edges denote the category-subcategory relation between the topics. Additionally, our recommender relies on the representation of any user's profile as a ranked subgraph of the knowledge graph, such that the nodes represent the interests of the specific users. For both recommendations, tweet and followee, our recommender uses the Steiner Tree of the user profile to calculate relevance. Our intuition behind using Steiner Tree is that given a set of topics, the Steiner Tree is the least cost-connected subgraph of the knowledge graph that contains these topics. If a concept connects two or more topics of interest in the knowledge graph, then this concept is likely itself a topic of interest. Although some topics may not be directly related to one's interests, we assume that if a concept belongs to the Steiner Tree of the topics of interest, then the likelihood of it being itself one increases.

1.1.2 Automatic generation of feature-agnostic datasets for fake news detection in social media

Algorithmic methods to automatically detect misleading content on social media are crucial to help mitigate fake news diffusion. Developing machine learning detection models is a data-driven process that raises the need for adequate and quality training datasets, including social media and fake news ground truth data. However, there are certain characteristics of this problem that make it uniquely challenging.

- The manual process of labeling news and post content is time-consuming and results in partial and outdated datasets.
- There is significant diversity among existing datasets that raises severe problems to the reliable assessment and comparison of the detection models.
- Most approaches focus on specific subsets of features that can be measured in multiple ways, while network propagation features are underestimated.
- The nature of fake news content is not uniform and is related to newly emerging, time-critical events, which existing knowledge bases and fact-checkers may not have properly verified yet.

To tackle these challenges, we developed the PHONY infrastructure, which automates the generation of datasets that contain fake news and their propagation footprints on Twitter. Specifically, the infrastructure makes use of the fake news stories provided by fact-checking services and produces datasets that include social media posts that refer to those fake news stories. This is accomplished by building an incremental tweet inverted index containing streaming tweets. This index consists of broadened tweets that comprise the tweet text and metadata and web data related to each tweet. The generation of a new dataset starts with collecting fake news items from fact-checking websites and then analyzing and using them as queries to the inverted index.

PHONY generates uniform, flexible, and up-to-date silver standard datasets while allowing users to choose the fact-checking source, time period, and fake news type they are interested in, and the central idea behind our approach is described in [102]. The generated datasets are feature-agnostic, hence datasets that do not contain metrics of specific classification features. This allows users to freely choose the features and measurements that better suit their classification and detection methods. PHONY datasets contain text, user, network, and propagation data, and, to the best of our knowledge, all features encountered in the literature can be directly extracted using simple scripts.

1.1.3 Knowledge-based Recommendations for Data Exploration

Over the last years, data growth, volume, and the complexity of available data have increased the need for efficient data exploration systems. Search data systems usually face the following problems: natural language ambiguity, entity recognition and linking efficiency, recognizing domain-dependent entities, keywords, and concepts. At the same time, the available datasets need to be accessible by regular users who are not familiar with databases or the type, structure, and contents of the data.

In the context of our thesis, we developed the KNOwDE system that focuses on assisting users who aim to extract knowledge from data without precisely knowing the specific data structure and contents. The system tackles the data exploration problem in the light of generating efficient knowledge-based and data-based recommendations and providing relevant data insights to the user. To provide recommendations for queries relevant to their interests and the data, we use knowledge bases and a graph derived from the database to explore. The semantically relevant recommendations are enriched using further recommendations based on generalization and specialization relations from the knowledge bases. Finally, KNOwDE, using graph theory algorithms, recommends data objects and provides two graph views based on user selections, offering a visualization that enables users to acquire a more comprehensive look at the results and helps them interact with the recommended objects and the objects related to them.

1.1.4 Applications leveraging social network data for exploring datasets

In the context of the thesis, we have developed some methods and services, which facilitate users that do not have any special knowledge on the data to evaluate, navigate and discover useful information in available datasets. These applications are the following:

1. **Calculation of Tweet-based attention for articles relevant to COVID-19**

   The outbreak of the coronavirus pandemic has turned us into new priorities regarding social network analytics. Specifically, we collaborated with the team developing the BIP!Finder [1] to help create a dataset that will consist

[1] https://bip.imsi.athenarc.gr/

of literature related to COVID-19 and will comprise a social media altmetric indicator. As we write these lines, pandemic-relevant articles are published rapidly, making it very difficult to explore and extract useful information from them effectively. In this context, the team's main objective was to produce BIP4COVID19, an openly available dataset containing various impact measures calculated for COVID-19-related literature [145].

In order to capture the popularity of the articles, we opted to measure the Twitter attention received by each article. This altmetric augments the citation-based metrics and involves measuring the number of recent tweets mentioning the scientific articles. We consider this a measure of social media attention for each literature article relevant to COVID-19.

However, accessing the entire stream of tweets or performing multiple searches on historical tweets would yield incomplete results due to limitations posed by the Twitter API. Thus, we used an existing dataset with tweets related to COVID-19 and mined them for URLs pointing to the articles in our database. Therefore, we produced the URLs of the articles in doi.org, PubMed, and PMC based on the corresponding identifiers. We periodically produce the URLs to measure the Twitter attention indicator based on the latest tweets, which we release on frequent version updates on Zenodo [143] and on Bip4Covid website [2].

2. **Combining Geolocated Social Data for Urban Area Profiling**

Location data have been widely used in event detection, sentiment analysis, hotspot identification, typical movement patterns identification, city maps enrichment, etc. However, the location data face some serious problems:

- Volunteered geographic information (VGI: extracted from social media, microblogging platforms, check-in applications, mobile phone GPS) contributed by online users is imprecise and inaccurate by design
- Commercial Point of Interest (PoI) information (offered by leading web providers like Google, Here, Bing, Foursquare) set limitations on the use of those APIs, thus providing users with a very locally-limited view of the existing city infrastructure that cannot be directly used to extract additional information for city-scale areas.
- Data from government agencies despite being official, curated, of excellent quality, and impossible to collect by individuals, it has the obvious disadvantage that it cannot be real-time, it is usually not available through APIs, and most importantly, it may be updated at very infrequent intervals (e.g., census data), therefore at risk of being rather outdated.

To assist regular users in exploring available data related to urban areas, we developed the CitySense framework. CitySense is a dynamic urban area viewer that integrates various datasets related to an urban area, providing a rich visualization of a city's life. Our work focused on combining disparate datasets of various origins to provide a more comprehensive picture of a geographical

[2]https://bip.covid19.athenarc.gr/

area. To accomplish that, we developed the CityProfiler, a subsystem responsible for data collection, and we created the CitySense Database that stores the diverse data. To enrich the PoI data, we designed CityProfiler to collect user-generated geolocated Twitter activity [70]. Moreover, we focused on efficiently spatial aggregating, visualizing, and presenting the end-user with a rich view of any data source so that users can easily interpret this information. We made CitySense available to users through the corresponding web application [3] that builds a unified view of our use-case, the urban area of Chicago, utilizing open data from administrative sources, online PoI APIs, and tweets.

1.2 Thesis contributions

- **Tweet and followee personalized recommendations based on knowledge graphs**

  We developed a new content-based method that uses knowledge graphs for (a) personalized tweet recommendations and (b) personalized followee recommendations. The tweet recommendations compose an alternative personalized timeline containing streaming tweets that strongly match the user's interests, and the personalized recommendations of followees is a ranked list of Twitter accounts with similar interests to the user. Both our methods:

  1. are based on the representation of user profiles as topics of interest (ToIs). In our context, ToIs are nodes of a predefined KG that represent the interests of specific users and that are connected in a least-cost way using the Steiner Tree algorithm
  2. can adapt to cover new topics of interest and reduce the effects of over-specialization and over-recommendation
  3. are not impaired by the limitations posed by Twitter concerning the availability of the user graph data

  We conducted two experiments: one to evaluate the tweet and followee recommender system and another for which we used a large dataset to evaluate our approach's efficiency and scalability. The efficiency of our method outperforms in many cases the state-of-the-art approaches, which we have implemented for evaluation purposes, and yields good results in terms of precision and time scalability.

- **Automatic generation of feature-agnostic datasets for fake news detection in social media**

  We developed PHONY, an infrastructure for automating the generation of feature-agnostic silver standard datasets. These datasets contain fake news and their propagation footprints in the Twitter network based on the fake news stories provided by curated fact-checking websites and comprise the necessary data to extract all features encountered in the literature, including network propagation and semantic features.

[3] http://geoprofiler.imsi.athenarc.gr/

Moreover, we explored the range of the misinformation and disinformation detection features encountered in the literature. This thesis provides a complete feature typology based on the analysis and systematization of all available features. This typology can be useful for training fake news detection models that cover all types of misinformation.

Furthermore, we used PHONY to generate the Greek PHONY Dataset, a large dataset of fake news propagating in the Greek Twittersphere. The efficiency of PHONY was measured by evaluating the Greek PHONY Dataset that reached an average precision of 77,5%.

- **Knowledge-based Recommendations for Data Exploration**

  We developed KNOwDE, a system for generating efficient knowledge-based and data-based recommendations for data exploration based on user search queries. KNOwDE's underlying method is that the concepts in the user query are matched with the concepts in the database with the help of knowledge bases that provide alternative keywords and key-phrases that are semantically related to the original ones to capture the actual context of the users' interests. Specifically, we use recommendations based on knowledge bases (KBs) and a Graph extracted from the database (Data Graph) to assist users unsure how to form a correct query. These recommendations help users form queries relevant to their interests and the database. The user can interact with the system by expanding the original query with these alternative keywords that are ranked by mining Keyword Graphs, graph representations for each keyword, and its related alternatives. Moreover, KNOwDE extends these alternatives and offers further recommendations based on generalization and specialization relations derived from the knowledge bases.

  Furthermore, KNOwDE utilizes the selected keywords and a graph derived from the database (Data Graph) to generate ranked data object recommendations based on frequency and PageRank. To provide a more comprehensive look at the results, we generate two Data Graph views based on the selected alternative keywords, namely the Subgraph S and the Steiner tree, that connect the most critical and associated database objects.

  We have implemented KNOwDE, which currently integrates the CORDIS[4] database and the DBPedia[5] and ConceptNet[6] knowledge bases. We have also deployed the KNOwDE[7] web application using a user-friendly interface.

- **Calculation of Tweet-based attention for articles relevant to COVID-19**

  We designed a methodology for calculating an attention altmetric indicator extracted from social media data for ranking literature related to COVID-19. To this end, we mined an existing dataset with tweets related to COVID-19 with a set of URLs pointing to the articles related to COVID-19. This altmetric, called Social Media Attention, was our contribution to the Bip4COVID[8]

[4]https://cordis.europa.eu/en
[5]https://wiki.dbpedia.org/
[6]https://conceptnet.io/
[7]http://knowde.imsi.athenarc.gr/
[8]https://zenodo.org/record/5560080

dataset. BIP4COVID19, is an openly available dataset containing various impact measures calculated for COVID-19-related literature and has gathered a lot of attention on Zenodo (151,570 unique views and 14,232 unique downloads so far).

- **Combining Geolocated Social Data for Urban Area Profiling**

  We have designed and developed CitySense framework, a dynamic urban area viewer that integrates various datasets related to an urban area, providing a rich visualization of a city's life. CitySense combines data from administrative sources, Point of Interest APIs, and the Twitter social network to provide a unified view of all available spatial information about an urban area. To this end, we developed the CityProfiler, a subsystem responsible for data collection and efficient spatial aggregation, and the CitySense Database that stores the collected data.

  Moreover, we developed the corresponding web application [9] that builds a unified view of our use-case, the urban area of Chicago. In this context, we focused on visualizing and presenting the end-user with a rich view of available data to easily interpret this information through a friendly user interface.

1.3 Thesis outline

The thesis is structured as follows. Chapter 2 introduces some background knowledge regarding the key concepts in the dissertation. In chapter 3, we give a detailed presentation of our method for generating personalized recommendations based on knowledge graphs. Chapter 4 presents PHONY Infrastructure, which automates the generation of datasets for fake news detection in Twitter. In chapter 5 we present KNOwDE, that provides knowledge-based recommendations for data exploration, and in chapter 6, we present two applications that leverage social network data for exploring datasets, Bip4COVID, and CitySense. We conclude the thesis in chapter 8 and record the future research paths.

[9] http://geoprofiler.imsi.athenarc.gr/

Chapter 2

Background concepts

2.1 Twitter social network: basics and terminology

Twitter is a microblogging and social networking service that allows users to send and read short messages known as "tweets", including photos, videos, and links.

According to recent social media facts and figures, Twitter currently stands as one of the leading social media globally. Moreover, in 2019, the company's revenue was calculated to be 3.46 billion U.S. dollars, while most revenues came from advertising. Figure 2.1 depicts some interesting facts and figures regarding the Twitter service extracted from Statista[1].

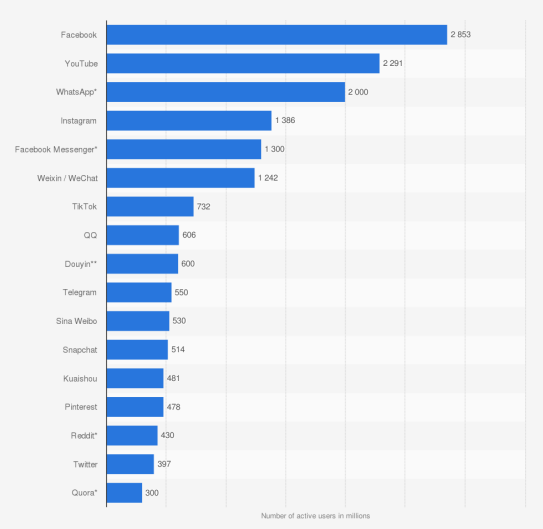
While following influencers remained an important use of the service, businesses and political campaigns discovered the value of Twitter as a communication tool and began using Twitter for promotions and events.

A remarkable step in the evolution of Twitter usage was its increased use as a journalism tool. More and more journalists, both amateurs and professionals, use Twitter to channel their news stories and points of view, shaping its profile into an up-to-date, online, live news source. Also, non-journalist user tweeting is becoming increasingly prominent during events like sport, tv, news events, establishing Twitter as an emerging outlet for the diffusion of information during events.

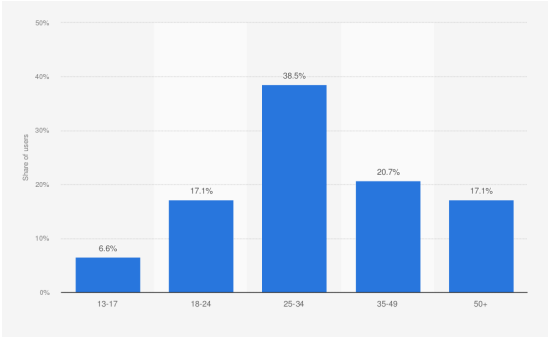
In this thesis, we used Twitter data in multiple contexts. Therefore, we need to define some specific terms that will be used throughout this thesis:

- **tweet**: is a short message (up to 280 characters) and may contain photos, GIFs, videos, and text.
- **timeline**: is a personalized "home page" containing a stream of Tweets shared by user friends and followees ranked by a recommendation algorithm.
- **follow**: is subscribing to a Twitter account.
- **follower**: is a user that subscribes to see the tweets of another user (**followee**) on his timeline.
- **retweet**: is a tweet that users forward to their followers.
- **reply**: is a response to another person's tweet.

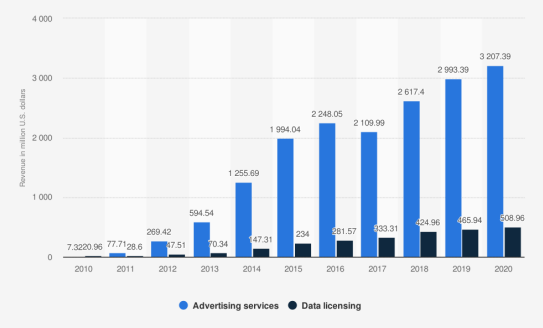
[1] https://www.statista.com/



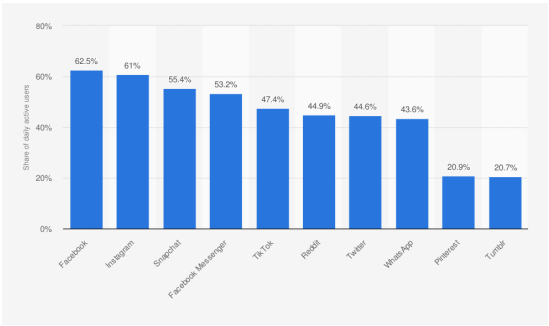
(a) Global social networks ranked by number of users in 2021 (in millions)



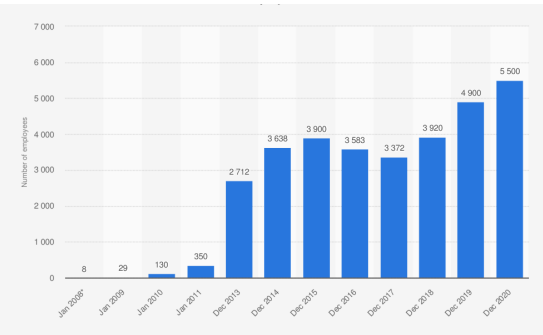
(b) Distribution of global audiences in 2021 by age group



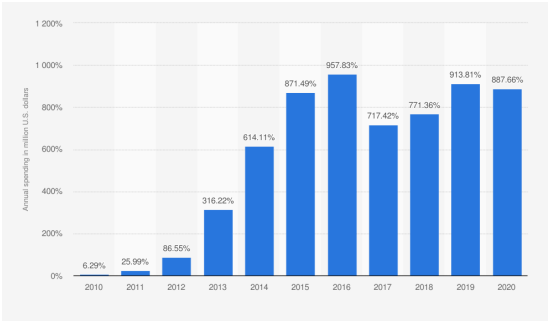
(c) Annual revenue from 2010-2020 by segment (in million USD)



(d) Daily engagement rate of US mobile social users in 2020



(e) Number of employees from 2008 to 2020



(f) Sales and marketing spending from 2010 to 2020

Figure 2.1: Twitter statistics

- **hashtag**: is any word immediately preceded by the symbol. When users click or tap on a hashtag, they see tweets containing the same keyword or topic.

- **geolocation, geotagging**: is information tagged on a tweet and containing the location of the user when she posted the tweet.

Table 2.1 comprises some milestones in the evolution of the Twitter recommen-

dation algorithm:

Table 2.1: Evolution of Twitter recommendation algorithms

Algorithm	Description	Year
Twitter Feed 1.0	Timeline displayed tweets in reverse chronological order	2006
Twitter recommendations	Timelines start to include recommended tweets, topics, and accounts	2014
While you were away	A forerunner to "In case you missed it," recapped select tweets, chosen based on "engagement and other factors"	2015
Reordered timelines	The first algorithmic restructuring of timelines that pushed the "best tweets" to the top	2016
Relevance model and ICYMI	Tweets were being scored on a relevance model that used recency, engagement, and interactions to personalize feeds. "While you were away" was also swapped for "In case you missed it" (ICYMI)	2017
Top Tweets vs. Latest Tweets	Allows users to toggle between Top Tweets and Latest Tweets, i.e. an algorithmic recommendation and a reverse chronological order timeline. Twitter revealed that it uses textitDeepBird, a deep learning system, to predict which tweets users will find interesting and appealing	2018
Customizable Timelines	Timelines can be swapped for up to five different lists	2019
Topics	Topics allow users to follow conversations, i.e. following a Topic adds related tweets, users, events, and ads to the timeline	2019
Fleets	Fleets are short videos that only stay online for 24 hours	2020
Liked By and Followed By	Twitter removed the visibility of Liked By and Followed By from showing up from people and topics that users don't already follow	2021

Twitter provides an API for developing apps for data collection (tweets, users, direct messages, lists, trends, media, places) and integrating with apps and systems. The Twitter API[2] can be used to programmatically retrieve tweets, get meaningful data insights, and post new tweets. However, it poses limits regarding the number of requests and responses. The maximum number of allowed requests is based on a time interval, some specified period, or time window, and the most common request limit interval is fifteen minutes.

2.2 Knowledge graphs: definitions and outline

Knowledge graphs have emerged for organizing structured knowledge and for integrating information extracted from multiple data sources. They prevail in the intersection of emerging research areas such as Data structures, Databases, Knowledge representation, Graph theory, Applications (Machine learning), etc.

Specifically, knowledge graphs are knowledge bases that use graph-structured data models to represent objects, concepts, events (nodes), and relations (edges).

[2]https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api

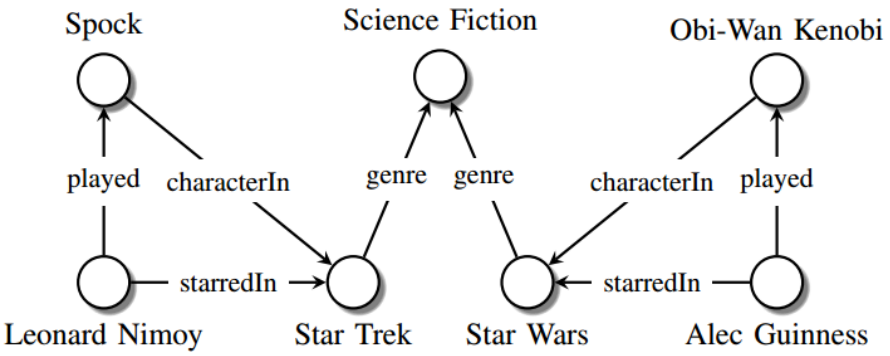


Figure 2.2: Example of knowledge graph

These edges represent semantic relations between the nodes. Recently, a large variety of definitions have been proposed [44]; still, both companies and open source communities have boosted the construction of both knowledge graphs and knowledge bases utilizing semantic analysis and extraction techniques. In this context, knowledge graphs are essential for turning human knowledge into manageable data for computer systems. Figure 2.2 depicts an example of knowledge graph [94]. Knowledge graphs have also started to play a key role in machine learning to combine global knowledge and explain learning outcomes.

Not every RDF graph is a knowledge graph since sometimes graph representation of data can be unnecessary to capture the semantic knowledge of the data. Also, not every knowledge base is a knowledge graph, since knowledge graphs require that entities are interlinked to one another.

Apart from organizing available knowledge on the Web, knowledge graphs have been used in various applications and industries: up-selling and cross-selling strategies through recommendations in retail, AI-based recommendations in entertainment, financial crime prevention, and investigation by understanding the flow of money across people and identifying suspicious transactions, organizing and categorizing medical research relationships for validating diagnoses and identifying treatment, customer relationships management by integrating external information (financial news, commercially sourced and curated data about supply chain relationships) with internal information about the same customer, etc.

Knowledge graphs play a significant role in developing Artificial Intelligence algorithms and systems. AI reasoning uses real-world representations. Hence, the representation quality influences both output storage and conclusion deriving processes for incorporating new knowledge.

Various techniques have been used for constructing, managing and exploiting knowledge graphs: link prediction for finding missing edges in knowledge graphs and recommending possible relations, named entity recognition and resolution, entity extraction and relation extraction comprise advanced NLP techniques, community detection/ entity clustering based on various similarity measures, visual question answering, graph embeddings for machine learning applications, etc.

Knowledge graphs are usually large and complex, while their generation is either crowdsourced or based on multiple sources. Some of the most important ongoing

knowledge graph projects are DBpedia[3], YAGO[4], Wikidata[5], ConceptNet[6], and NELL[7]. Wikidata is the most suitable source for person-related data, containing twice as many instances as DBpedia or YAGO, while organizations like companies are best described in YAGO. Additionally, DBpedia contains more places, including almost four times more cities. Although DBpedia and YAGO contain much more countries than Wikidata, Wikidata contains the most detailed information about countries and is available in multiple languages. YAGO is the most suitable source for events, both in terms of coverage and level of detail. Moreover, although NELL contains the largest number of chemical substances, the highest level of degree for chemicals is provided by Wikidata. Finally, YAGO contains the most significant number of astronomical objects.

[3]https://www.dbpedia.org/
[4]https://yago-knowledge.org/
[5]https://www.wikidata.org/wiki/Wikidata:Main$_{P}age$
[6]https://conceptnet.io/
[7]http://rtw.ml.cmu.edu/rtw/kbbrowser/

Chapter 3

Tweet and followee personalized recommendations based on knowledge graphs

3.1 Introduction

Twitter is one of the biggest and better-known microblogging sites, allowing its users to send and read short messages. Unlike other social media, the possible relationships among Twitter users are two, followee or follower. When a user publishes a tweet, it automatically appears on his home page and on the home pages of his followers (user home pages are also referred to as timelines).

Twitter is growing rapidly into one of the most popular social network services. Recent statistics show that more than 550 million users generate more than 300 million tweets every day, leading to a vast amount of information that can be made readily available and introducing new problems. Let us think of a user's homepage: it is growing every time a followee tweets; however, not all tweets are of interest. Because of this tweet overload, users are often tired of browsing tweets on their homepage, and as a result, they miss interesting tweets. A solution to this problem is an efficient recommender that helps users filter out uninteresting tweets and avoid cross-passing the interesting ones. Moreover, a similar recommender could recommend followees with similar or relevant interests, resulting in a more interesting timeline.

On a different front, knowledge engineers organize human knowledge in an objective way using semantic technologies. In this context, knowledge graphs (KGs) [45] are an instrumental tool for encoding the domains of human knowledge and the relations between them in a manageable way for computer systems. It is no coincidence that services like Google [1], Microsoft [2], and IBM [3] promote the development of such structures to improve their commercial services (navigation, search, personalized services, targeted advertising) and to increase the engagement of users [50]. It is our view that the semantic features of KGs make them suitable for analyzing social network data.

This chapter presents a content-based method [72] that uses KGs for (a) per-

[1] https://developers.google.com/knowledge-graph
[2] https://concept.research.microsoft.com
[3] https://www.ibm.com/watson

sonalized tweet recommendation [71] and (b) personalized followee recommendation [69]. Our method provides an alternative user "timeline" containing tweets that strongly match her interests. Note that these tweets may have been posted by users that are not her followees. This way, she will not miss interesting messages, even if they are posted by people that she does not follow; at the same time, irrelevant tweets are filtered out. Furthermore, our method for personalized followee recommendation helps users discover and follow people with similar interests. Both our methods are based on the representation of user-profiles as topics of interest (ToIs). In our context, ToIs are nodes of a predefined KG that represent the interests of specific users. Both our recommenders can adapt to cover new topics of interest and reduce the effects of over-specialization and over-recommendation. As another advantage, our method is not impaired by the limitations posed by Twitter concerning the availability of the user graph data. We implemented from scratch the best-known approaches in order to compare with them. The efficiency of our method outperforms in many cases the state-of-the-art approaches and yields good results in terms of precision and time scalability.

3.2 Related Work

A wide variety of tweet and followee recommendation methods can be found in the literature. These methods can be grouped into Collaborative Filtering, Content-Based, and Tweet Ranking methods. In what follows, we examine each of those categories.

3.2.1 Collaborative filtering

Collaborative filtering (CF) methods use the community data to build user profiles [132]. The intuition behind these methods is that users who share the same opinion on some topics (interesting, not interesting) tend to have the same opinion on other topics (user-based CF). Moreover, topics that produce the same opinion from some users tend to receive similar opinions from other users (item-based CF) [9, 120]. Both neighborhood-based methods [118, 127] and model-based methods [77, 111] are subcategories of CF used widely in tweet recommendation. Neighborhood-based methods recommend items based on the similarity of the user of the item neighbors, and model-based methods perform recommendations using matrix factorization model or the probabilistic latent factor model. CF methods use user graph data extracted from Twitter, like follow and retweet links, to construct a network structure. These methods [113, 163] apply network analysis algorithms to the network structure to find interesting messages.

However, the network construction requires a large volume of link data to be retrieved, stored, and analyzed and thus cannot be updated in an effective and scalable way when new tweets are published in the stream. Related works suggest several algorithms and network features [79]. Such a feature is the topology of the followers' network [19] used to recommend users. Collaborative filtering approaches require each tweet to get instant feedback from numerous users before being recommended to other users, known as the "cold-start" problem. In [111], authors use a model-based method, which proposes online update rules on a stochastic gradient descent style based on the last example observed. In [41], authors propose the RMFO-RSV

method that maintains a reservoir with a representative set of previously seen data points from the stream, which provides a significant boost in performance compared to the one obtained when only the last example is considered. In [62], the authors use Co-Factorization Machines (CoFM) to address the problem of simultaneously predicting user decisions and modeling content in social media by analyzing rich information gathered from Twitter. These methods consider the relationship between tweets and users and the relationship between users and publishers separately. The problem is that two tweets with the exact same text posted by two users will be evaluated differently, although they have the same content; thus, they are of the same interest to the user! In [150], the authors make recommendations with social trust information based on matrix factorization methods.

In [165], the authors present an extending topology-based algorithm for recommending users in Twitter. The proposed algorithm classifies the users according to their friendship relations and constructs a class including user ids to recommend the target user. User actions and user mentions are also used to optimize the results. In [29], a collaborative ranking model is proposed, CTR, which considers three major elements on Twitter: tweet topic level factors, user social relation factors, and explicit features such as the authority of the publisher and quality of the tweet. In [74], the authors propose a probabilistic model based on Probabilistic Latent Semantic Analysis (PLSA) to recommend potential followers to users on Twitter. In [22], the authors propose a methodology to infer interests using some user followees (topical experts) and social annotations (collected via the Twitter Lists feature). In [83], the authors provide followee recommendations by calculating user relevance scores, using neural networks to combine network topology and content of tweets. In [112], the authors recommend followees using a fuzzy system that exploits followee similarity along with text similarities. Finally, in [125] authors present GraphJet, a recently deployed system for real-time content recommendations in Twitter, based on a real-time bipartite interaction graph between users and tweets.

3.2.2 Content based

A standard solution to the cold start and complexity problems is to use other information like the textual content of the items to be recommended [21, 139]. In [14], the authors used crowdsourcing to categorize a set of tweets as interesting or uninteresting and reported that the presence of a URL link is a single, highly effective feature for selecting interesting tweets with more than 80% accuracy. However, this rule may incorrectly categorize an uninteresting tweet (links to meaningless content) as interesting. Content-based methods build profiles by using the user history tweets. Such recommenders are often used in domains where a large amount of textual content is available for each user, such as websites. Recommending interesting tweets using content is not easy because tweets are limited in size. Previous works in content-based methods mainly recommend tweets to users using content analysis like Latent Dirichlet Allocation (LDA) or TF IDF metrics to represent user interests. In [100], the authors first created bag-of-word profiles for individuals from their activities and then chose websites most relevant to the profile of the individual as recommendations. In [73, 152], the authors conducted topic modeling of temporally-sequenced Twitter documents and tried to model the topics over time continuously. These approaches learn topic shifts based on word distributions of tweets, while

TS-LDA in [164] the model is learning changes based on topic distributions. Another approach, the Labeled-LDA [108], is used to model a tweet using its labeled information and then built the probability distribution vector of latent topics to represent the content of tweets. Based on similarity between the topic vectors, the incoming tweets are marked as interesting or not interesting.

In [86], the authors used Explicit Semantic Analysis [52] to construct the user interest profile based on Wikipedia concepts, to re-rank his timeline. However, in Twitter, the content of user tweets is much limited and sparse, so that these explicit terms extracted from history tweets are insufficient to reflect user interests. For example, some latent interests or preferences cannot be characterized in content-based methods [77]. Another approach to analyzing Twitter that uses topics is TwitterRank, which aims to identify influential micro-bloggers [154]. This approach leverages LDA by creating a single document from all user Tweets and then discovering the topics by running LDA over this "document." Again, such an approach has LDA problems since the Twitter data is sparse, and the generated topics are based on terms rather than concepts. Most of the time, the Twitter activity of a user is insufficient for creating a reliable profile.

For this reason, a wide variety of approaches make use of both Content-Based and Collaborative Filtering methods. In [21], the authors proposed to create user profiles not from the contents of tweets of an individual but a group of tweets posted by related users. In [59], the authors evaluated a range of different profiling and recommendation strategies, based on a large dataset of Twitter users and their tweets and the relationships between them to make useful followee recommendations. [46] presents TRUPI, a system that combines the user social features and interactions and the history of her tweets and captures the dynamic level of user interests in different topics to accommodate the change of interests over time. In [92], the authors propose a method to predict the probability of a tweet being retweeted based on content features alone. In [64], the authors propose Ontology-Based recommendations for news recommendations, using a traditional term-based recommender and several semantic-based recommendation algorithms to compare unread news items with the user profile and recommending items with the highest similarity with the user. [135] proposes a semantic TF IDF method, which weighs each message according to two factors: TF-IDF and a semantic similarity measure. In [133], the authors provide real-time recommendations by building a graph of words. In [167], the authors represent users by user-topics LDA distribution and recommend the top-k similar users by computing hashtag frequencies.

3.2.3 Tweet ranking

Some recent approaches focus on recommending tweets from the user timeline. In [43], the authors use a learning-to-rank algorithm that uses content relevance, account authority, and tweet-specific features to rank the tweets in the timeline. Other approaches construct a tweet ranking model making use of the retweeting behavior of a user. For example, they rank both the tweets and the users based on their likelihood of getting a tweet retweeted [140].

The amount of information provided by Twitter is so large that most of the already mentioned algorithms become intractable. Many optimization methods were developed to reduce time complexity. For example, in [119], the authors applied

clustering algorithms to partition user populations, built neighborhoods for users from the partition, and considered only those neighborhoods when computing recommendations.

3.3 Relation to previous works

Our method avoids the efficiency problems of LDA or TF-IDF-based methods caused by the limited size of tweets. Our approach uses each tweet separately (assigns a topic to each tweet) compared to most of the content-based methods, which merge tweets and therefore defy the proper granularity for topic extraction. Moreover, our method takes advantage of the KG to recommend tweets of related topics, while other methods recommend the exact topics found.

In contrast to collaborative filtering techniques, our approach does not face the problem of resource availability since it makes no use of the Twitter user graph data. This problem is discussed in detail in Sect. 3.2. Whereas some other approaches require a lot of processing time, the overall time needed for our method to construct a new user timeline is minimum, and thus it can be implemented as an online streaming service. Finally, compared with Ontology-based recommenders, we believe that KGs are less complicated and provide a stable but also lightweight basis for tweet recommendations.

Figure 3.1: Knowledge Graph

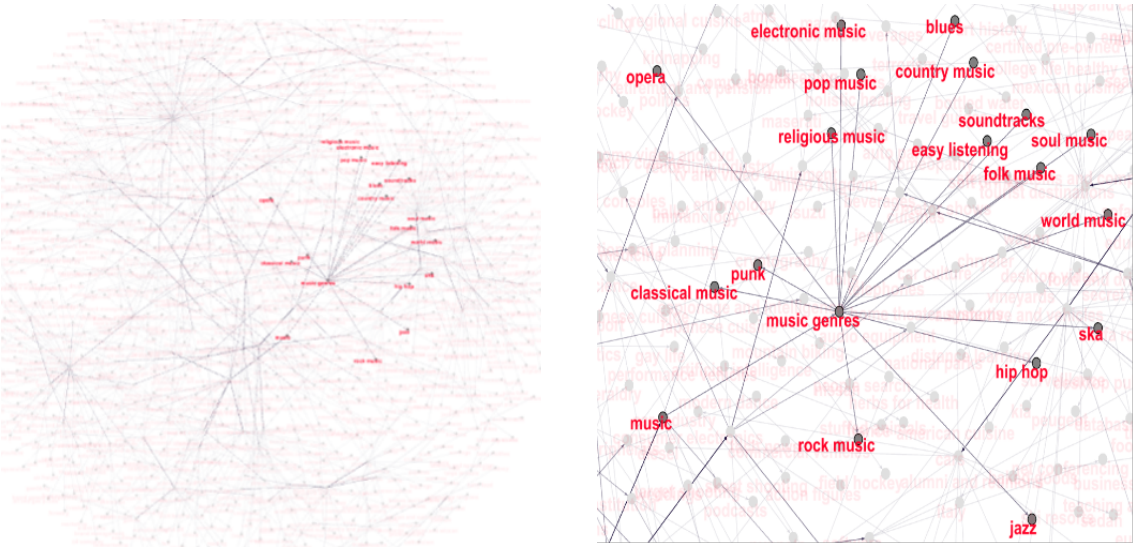


Figure 3.2: Music Genres

3.4 Overview

Our approach is based on two principles:

1. The representation of all possible user interests as a hierarchical KG, with more general concepts on top and more specific concepts as children. Each node corresponds to a ToI, while edges denote the category-subcategory relation between ToIs.

   To construct the KG, we opted to use the AlchemyAPI Taxonomy service and its Categories dataset, which is a set of concept categories and subcategories extending up to 5 levels deep. For example, the category "music genres", which is a subcategory of "music", has 17 subcategories, and each of them represents a music genre. This example is shown in figures 3.1 and 3.2. Alchemy Taxonomy concepts form a graph $G = (V, E)$, where $V = v_i$ is the concept set from AlchemyAPI Taxonomy. Each concept v_i is connected with the concept v_j, if and only if these concepts are related in AlchemyAPI Taxonomy Dataset via an edge that belongs to the set $E = e_i$ of graph edges. All edges are of equal weight. The KG consists of 1092 nodes (concepts) and 1323 edges (concept relations). We use the relation category-subcategory, and the existence of an edge between two concepts is an indicator of semantic relevance. The KG covers the vast majority of concepts that are used in everyday life, therefore it provides a wide knowledge base for our recommender.

2. The representation of the profile of any user as a subgraph of the KG, such that the nodes represent the interests of the specific users (ToIs). Those ToIs are subsequently ranked from the more specific towards the more general.

   The construction of user profiles requires extracting a subgraph of the KG, containing the user ToIs. The optimum such subgraph is extracted using the Steiner Tree [54] extraction algorithm. Given a graph $G = (V, E)$ and a set $R \in V$, a Steiner Tree is the least-cost connected subgraph spanning R. In our method, R contains the set of ToIs extracted from the user timeline, and all

edges are of equal length. To compute Steiner trees, we applied the function supplied by the Networkx python library. The goal of this approximation to the minimum Steiner tree is to extract a tree connecting all these ToIs with a minimal sum of costs along its edges. Finally, these ToIs are ranked to avoid recommendations based on very abstract ToIs using a DFS Postorder traversal of the tree.

3.5 Motivation

A pivotal issue for a recommender is selecting the criteria based on which the system will provide recommendations. Most content-based recommenders, which rely on text/keyword similarity, lack efficiency due to the small size of tweets. Unlike these methods, our approach is based on the semantic relevance between the user interests and the incoming tweets. Choosing semantic relevance as the recommendation criterion has the following advantages. Considering that users read and write content over multiple topics, our intuition is that a recommender should consider the conceptual associations between these topics, which are—up to some degree—objective and immutable.

Our method is based on the representation of user-profiles as Topics of Interest (ToIs). Specifically, profiles are constructed upon a predefined structure, the Knowledge Graph. The KG consists of nodes representing concepts and objects (e.g., events, persons, entities, locations that are potential ToIs) and edges representing relations between them. In our context, ToIs are nodes of a KG, which represent the interests of specific users. The usage of a KG provides us with a common basis for (a) generating user interest profiles and (b) calculating the semantic relevance between them on the one hand and the incoming tweets on the other. Moreover, KG semantically outperforms the LDA self-topic approaches and term frequency approaches, whose efficiency suffers due to the small size of tweets.

A common problem for content-based recommenders is over-specialization. Content-based recommenders suggest items whose similarity scores are high when matched against a user profile. Such approaches, however, restrict the user exclusively to tweets very similar to those already seen by providing recommendations that contain recurrent information and certainly not covering one's range of interests. This problem, called over-specialization, is avoided by our recommender. The intuition is that our recommender provides content covering relevant ToIs and ensures that the recommended tweets span "as much as possible" on the KG and do not come all of the same node/ToI. To achieve that, we extend the user profile with related ToIs from the KG. The KG contains thousands of nodes, so this extension should respect some constraints. For example, if a node connects two or more user ToIs in the KG, this node is likely itself a ToI. Continuing on this line, the connection of user ToIs in an optimum way leads to a broader profile, exploiting the semantic relations between ToIs. We use the Steiner Tree algorithm to accomplish this optimum connection, as discussed in the previous section.

Depending on the type of KG, nodes can represent either exclusively specific objects (events, entities, persons, etc.) or a combination of categories (concepts) and objects. In our approach, as we show in the next section, the KG is a topic taxonomy, including topic categories and objects. The recommendation based on a very abstract category results in too many possibly not interesting recommenda-

tions. This problem is known as over-reccommendation. For example, let us assume a user is interested in ancient history. A recommender of general topics would recommend tweets about "science", a super-category of history that includes chemistry, computer science, medicine, etc. Ignoring these abstract nodes during user profiling is impossible since the hierarchy structure of our KG does not allow us to find an optimum subgraph that does not include them. Our approach avoids this effect by ranking the user profile, as shown in section 3.8.2.

3.6 Followee recommendation

Apart from recommending interesting tweets, another way to improve the user timeline is to recommend followees with similar or relevant interests. We assume that users choose whom to follow based on certain criteria, from which the most important is whether the followee posts tweets that are interesting to the user. For example, a person interested in sports and politics is likely to follow a person interested in these topics. However, apart from some famous individuals, a regular Twitter user is not widely able to know who shares her interests. Using the same underlying ideas (namely the KG and the Steiner tree) to profile users, we introduce a followee recommendation method that uses a similarity metric, called InterSim, discussed in Sect. 4.4. Additionally, our method can be applied using any KG since the specific tools and taxonomies do not restrict our method's basic principles.

3.7 Resource availability

Another essential advantage of our approach is that we avoid the problem of resource availability. This common problem that many recommenders must face is caused by the cost of the resources (Twitter data) necessary for the profiling. The Twitter API poses restrictions regarding the user graph data significantly greater than those regarding timeline data (tweets). For example, a recommender can request only 15 friends or followers of a specific account every 15 min, while the restriction for requesting the account's tweets is 1500 every 15 min. Our approach makes no use of Twitter user graph data (friends and followers' relations information) and therefore avoids the problem of resource availability. Instead, we use a set of the user's most recent tweets, which is automatically updated at fixed time intervals. This way, our recommender can dynamically adapt to cover new topics of interest that may arise. At the same time, our method allows for a lightweight implementation.

3.8 Tweet and followee recommendation model

In this section, we present in detail our recommendation model, which consists of a common user profiling process and two distinct recommenders: the tweet recommender and the followee recommender. These recommenders provide recommendation lists of tweets and followees, respectively.

Figure 3.3 depicts the overall architecture of our recommendation model, which consists of the tweet representation unit, the user profiling unit, the followee recommender, and the tweet recommender. Those are presented in detail in the following sections.

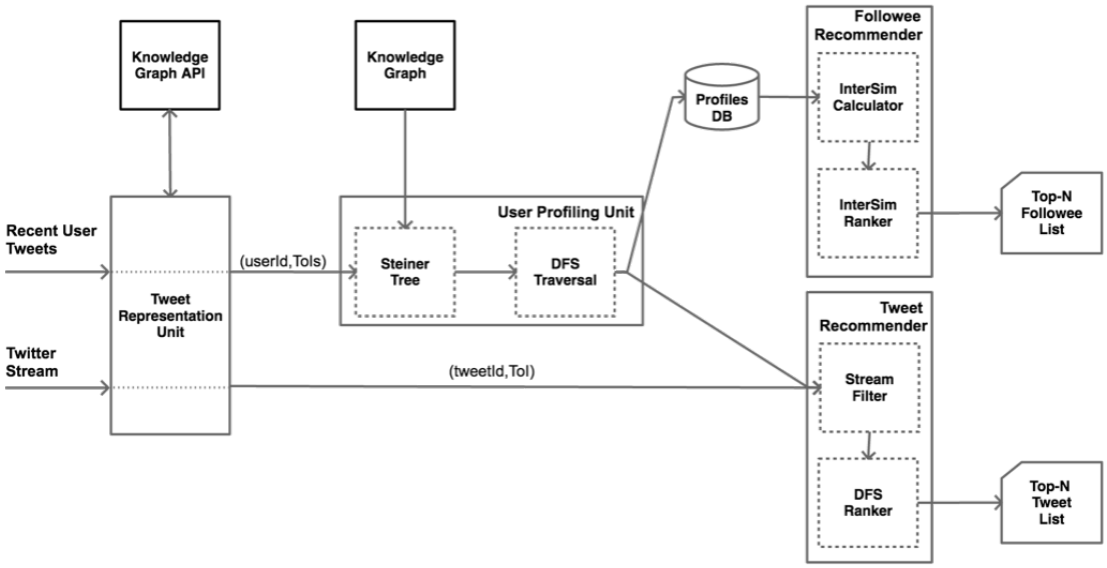


Figure 3.3: Tweet and followee recommendations based on knowledge graphs

3.8.1 Tweet representation unit

The function of this unit is to assign ToIs to tweets. The unit receives as input (a) a set of user tweets and (b) the streaming tweets from Twitter. The output is (a) a list of ToIs for each user and (b) a ToI for each tweet. First, every tweet is pre-processed to remove special characters (emoticons, etc.) and expand shortened URLs. Then, the unit assigns a ToI from the predefined KG to each tweet, using AlchemyAPI's Taxonomy API. The Taxonomy API is an online service for semantic text analysis that assigns concepts from the AlchemyAPI Taxonomy Categories dataset (described in Section 3.4) to tweets. This service automatically categorizes text and HTML into its most likely topic category from the KG.

3.8.2 User profiling unit

This unit is responsible for constructing user profiles. The unit receives as input a list of ToIs for each user, which is provided by the Tweet Representation unit. The output is an extended ToI list for each user. The unit extends the ToI list by finding related ToIs from the KG using the semantic relations between them. Specifically, this extended list is extracted from the KG using the Steiner Tree algorithm (Section 3.5). The intuition behind using Steiner Tree is as follows. Given a set of ToIs, Steiner Tree is the least cost-connected subgraph of KG that contains these ToIs. If a concept connects two or more ToIs in the KG, then this concept is likely itself a topic of interest. Although some topics may not be directly related to one's interests, we assume that if a concept belongs to the Steiner Tree of the ToIs, then the likelihood of it being itself ToI increases.

Next, the extended list is ranked to avoid recommending tweets and followees based on very abstract ToIs. As we discussed in Section 3.5, we assume that a user

is interested in reading tweets about specific topics. Therefore, our method ranks the user profile using a DFS Post Order Traversal, whose main effect is to explore deeper into the graph, hence promote more specific topics. This traversal requires a root node, which should be of the most abstract level in our KG. The User Profiling unit uses as root node the node of the most abstract level, which is closest to the node/ToI found the most frequent in the tweets of the user. This ranking method is named TGS-post. Finally, user profiles are stored in a database. A complete profiling example is presented in Section 3.9.

We designed and implemented two alternative ranking methods:

- TGS-tf: The ToIs are ranked based on the frequency of their occurrences in user tweets.
- TGS-bfs: The ToIs are ranked based on the BFS traversal of the Steiner Tree.

3.8.3 Tweet and followee recommenders

Given the ranked user profiles generated by the User Profiling unit, we now focus on the explanation of the two recommenders:

- The Tweet Recommender receives as input a ToI for each streaming tweet, as assigned by the Tweet Representation module. If a streaming tweet is assigned to a topic included in the user profile, then the Stream Filter temporarily stores the tweet in a database. The DFS Ranker will then rank the tweets based on the order of ToIs in the user profile (DFS) and store them in the Top-N Tweet List.
- The Followee Recommender calculates user interest similarity and ranks the stored user profiles. Specifically, for each user profile stored in the database, the InterSim Calculator determines the Interest Similarity (InterSim) with all other users. InterSim is calculated by measuring the subgraph overlap between these profiles. Thus, the number of common ToIs between a user and every other profile stored in the database gives us the user interest similarity. Moreover, we use the profile graph diameter as a normalization factor for each calculation. Finally, the InterSim Ranker ranks the profiles in descending order of interest and stores them in the Top-N Followee List.

3.9 A concrete example

As an example, consider Paul Mason, a widely known journalist and broadcaster who currently works as economics editor at Channel 4 News in the UK. We retrieve his timeline, and the Tweet Representation unit assigns a ToI to each tweet. The tweets and the corresponding ToIs are shown in Table 3.1.

The list of assigned ToIs is: annual report, politics, radio, reading, lobbying, government (three instances), tech news, business and industrial, elections, unions.

Next, the User Profiling unit applies the Steiner Tree using as input this list of ToIs. The resulting Steiner Tree is depicted in Figure 3.4. It consists of the following ToIs: *annual report, radio, business and industrial, company, art and entertainment, hobbies and interests, reading, law, govt and politics, government, politics, society,*

Tweets	Topics of Interest
“Reglingnoics”! https://t.co/q5lozpnd4l in other respects “had so many strange ideas”! Greekdept	Annual report
According to Regling I had “strange ideas” http://t.co/Qwp4ww8d9 Read them here ”compare them to Mr Regling’s”	Politics
Tonight on BBC One’s Question Time http://t.co/wghmmx4grh	Radio
Il Fatto Quotidiano interview on M. Renzi’s ‘comment’ ”my reply” http://t.co/fhfmidntzd	Reading
Interviewed by POLITICO on the 3rd Bailout, Schauble-Merkel, the Eurozone ”the refugee crisis” http://t.co/hpuyrmvggw	Lobbying
DER SPIEGEL: Complicit in Corruption: How German Companies Bribed Their Way to Greek Deals	Government
Ne pas manquer vendredi soir le prochain live de Mediapart: deux heures avec https://t.co/xtktgivrba	Tech news
fThe lenders are the real winners in Greece http://t.co/ltjtyvu5zw	Government
We would like to see you again fighting together for Syriza victory...Your division it’s a defea...	Business and industrial
The double purpose of these elections http://t.co/phl26kcmf2	Elections
Paul Krugman on Greece’s 3rd mou: an agreement designed to fail http://t.co/hhwiw5fgop	Government
Yanis Varoufakis—‘Left should beware of friends who fear confronting the rich’ http://t.co/posnmlqviq	Unions

Table 3.1: Assignment of Paul Mason’s Tweets

work, unions, technology and computing, tech news, elections, lobbying. Afterward, the unit ranks these ToIs using a DFS post-order traversal of the tree. As described in Section 3.8.2, the root node is chosen based on the frequency of the ToIs in the tweets of the user. In this example, the most frequent ToI is the topic ”government”. Thus, according to Alchemy Taxonomy, the nearest abstract ToI in the Steiner Tree is the topic ”law, govt and politics”, chosen as the root node.

The final ranked profile of Paul Mason is: *government, elections, lobbying, politics, annual report, company, tech news, reading, unions, work, society, radio, art and entertainment, hobbies and interests, business and industrial, law, govt and politics.*

Next, the tweet recommender receives the streaming tweets and the corresponding ToIs and filters those included in the user profile. For example, let us assume the following ToIs for a stream of tweets:

- ToI of tweet1: radio
- ToI of tweet2: statistics
- ToI of tweet3: elections

First, the Tweet Recommender filters out tweet2 since ”statistics” does not belong to the user profile. Then, the Tweet Recommender promotes tweet3 over tweet2,

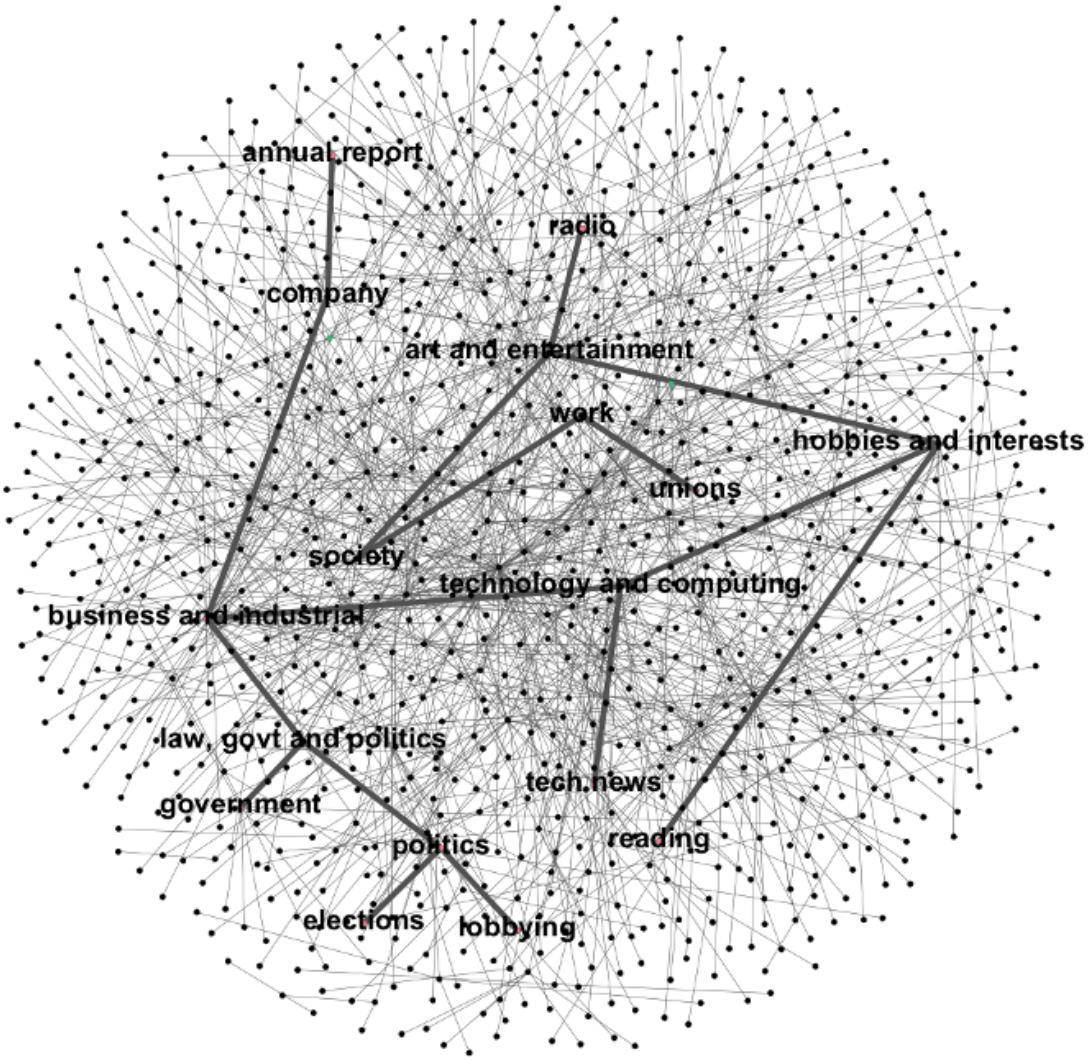


Figure 3.4: Paul Mason's profile

because "elections" has higher priority than "radio" according to the ranked user profile. Hence, the recommendation list is (1) tweet3 and (2) tweet1.

For a followee recommendation example, we picked two random Twitter users, user1 and user2, with equal-sized profiles and profiled them. The KG's nodes representing the overlap between Mason and user1 are surrounded by rectangles and the common nodes between Mason and user2 by circles, as shown in Figure 3.5. Thus, InterSim(Mason, user1) = 4 and InterSim(Mason, user2) = 3, which means that user1 will take a higher place in the followee recommendation list.

3.10 Experimental evaluation

To evaluate our tweet and followee recommenders, we conducted two offline evaluation tests presented in Sections 3.10.1 and 3.10.2. We compared the results of the tweet recommendation method with the most popular state-of-the-art methods. Furthermore, we conducted a large-scale, in-depth evaluation test using a large real-life dataset along with a runtime test of the proposed method. We present the

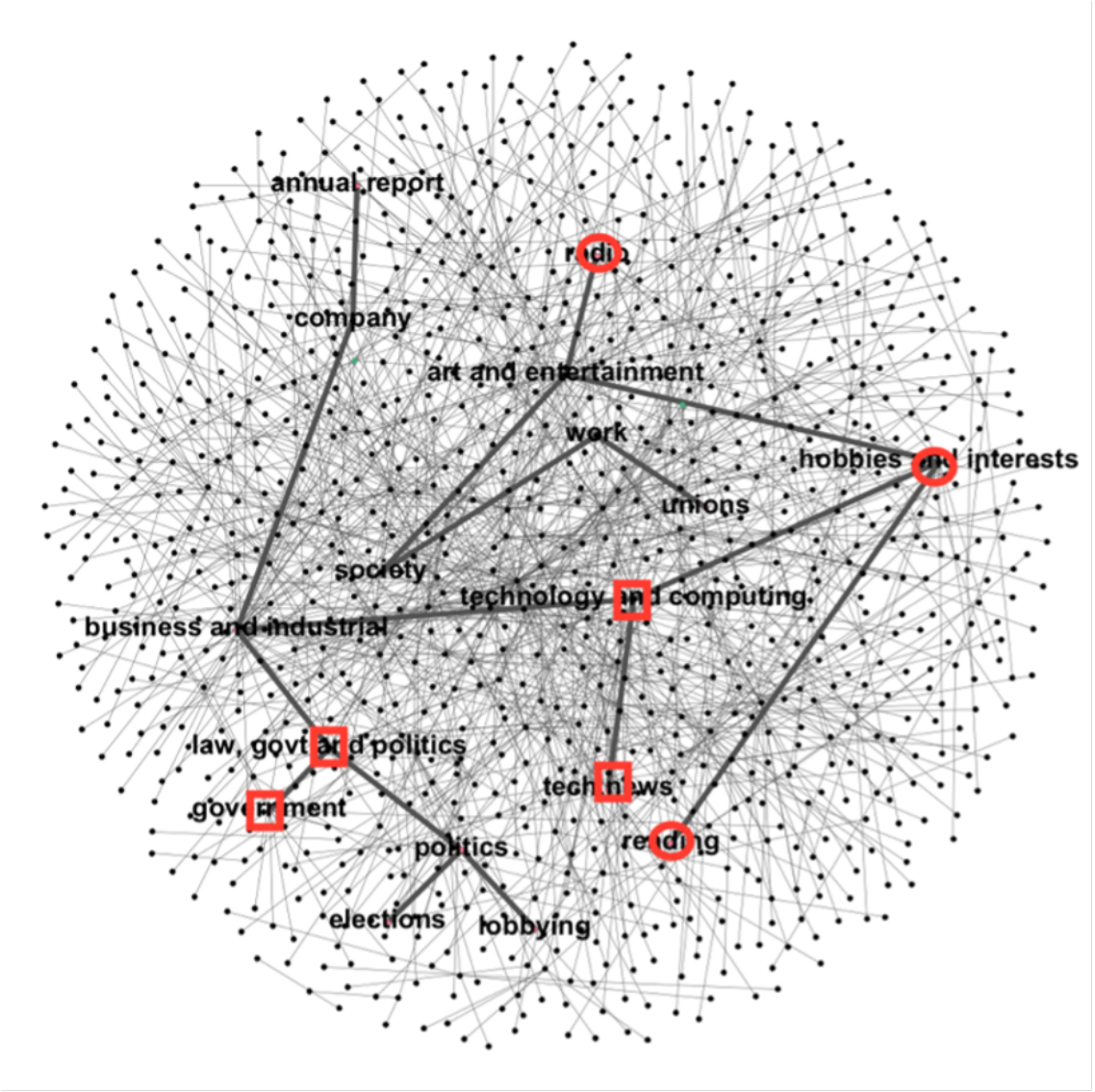


Figure 3.5: Profile overlap of users

results in Section 3.10.3, proving our method's stability and runtime scalability.

3.10.1 Comparing the tweet recommender with other approaches

We conducted an offline evaluation test to evaluate our tweet recommender and compared it with the most popular state-of-the-art methods. For a set of users, we gathered their most recent tweets, constructed their profiles, and recommended tweets based on the approach described in Section 3.8. We constructed the user dataset by crawling tweets and retweets from "The Twitter 100" users of 2012 [4]. This is a list of Britain's most influential users of 2012 based on PeerIndex that measures interactions across the web to help users understand their impact in social media. First, we constructed user profiles as follows:

[4]The Twitter 100: Britain's titans of the Twittersphere list - https://www.independent.co.uk/news/people/news/twitter-100-britain-s-titans-twittersphere-7466850.html

- Crawl twelve most recent tweets of the user (twelve was chosen to avoid scalability and info availability issues, due to Twitter API Rate limits).
- Assign a topic (ToI) to every tweet, as described in Section 3.8.1.
- Extract the Steiner Tree from the Knowledge Graph containing the ToIs, as described in Section 3.8.2.
- Execute a DFS traversal of the Steiner Tree as described in Section 3.8.2.

The resulting tree is the user profile. Finally, our dataset consists of a hundred users and their profiles, which are represented as vectors of ToIs ordered according to the DFS traversal. Subsequently, we constructed a test set to evaluate our method. We decided to build this test dataset out of the users' retweets because we assume that when a user retweets a post, he is most likely interested in it. Then we assigned a ToI to each retweet, as described in Section 3.8.1.

The test dataset consists of the most recent retweets crawled from the timelines of the users (500 retweets) and their Topics of Interest. The test process was made in the following stages for each user in the first dataset:

- Get user profile.
- For each ToI in the vector (beginning from the first) get all retweets of the same ToI from the test dataset.
- Store them in the recommendation list.
- Continue from stage 2 for the next ToI in the profile vector.

k	**Precision-at-k**
1	0.236559139785
2	0.195652173913
3	0.195652173913
4	0.16847826087
5	0.154347826087
6	0.143115942029
7	0.143115942029
8	0.126358695652
9	0.115942028986
10	0.105434782609

Table 3.2: Tweet Recommendations Precision-at-k

Mean average precision (MAP@10)	0.157973978161
Overall accuracy	0.988854305118

Table 3.3: Tweet Recommendations Precision and Accuracy

This way, we manage to rank all retweets from the test dataset according to profile and store them in the recommendation list. Subsequently, we computed three performance measures: precision-at-k, mean average precision, and overall

accuracy. Precision-at-k corresponds to the precision (information retrieval performance measure) calculated in the first k recommendations in the recommendation list. As relevant elements, we consider the retweets made by the user. We conducted experiments from k = 1 to 10. The results are shown in Tables 3.2 and 3.3.

Figure 3.6a depicts the mean average precision comparison between our method, TGS-post, our alternative rankings, TGS-tf and TGS-bfs, and four state-of-the-art approaches. Most approaches do not provide detailed instructions regarding their implementation. For the comparison to be fair, we implemented four popular methods, namely a simple LDA (lda), two TF-IDF (tfsimple and tfpairs, i.e., single word and word pairs), and a collaborative filtering approach (muifuot [101]). As we can see, our method reaches a mean average precision score of 15.7% and outperforms the other methods.

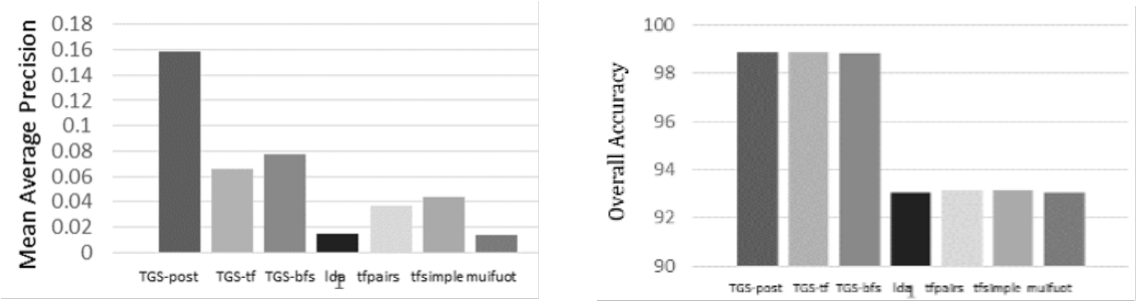
Figure 3.6b depicts the overall accuracy comparison between these methods. As we can see, our method reaches an overall accuracy score of 98.8% and outperforms the state-of-the-art methods.

Both efficiency metrics show that our recommender can successfully retrieve the interesting (retweeted) and not interesting tweets (true positive and true negative results), but still recommends some tweets that were not retweeted by the user (false positives). We can also observe that our model outperforms in terms of precision and accuracy the most common state-of-the-art methods (lda, tfidf-simple, tfidf-word pairs, muifuot) mentioned in Section 3.2.

3.10.2 Followee recommendation experiment

In order to evaluate our followee recommender, we conducted an offline evaluation test based on the dataset presented in Section 3.10.1. First, we constructed user interest profiles as in Section 3.10.1. Subsequently, we crawled the followees of all 100 users and used it as ground truth to compare with our recommender results. Finally, we used this dataset as input to our recommender. The evaluation process was made in the following stages for each user in the dataset:

- Calculate Interest Similarity (InterSim), as described in Section 3.8.3, between the user and every other user in the dataset.
- Rank recommended users in descending order of Inter-Sim
- Store top-k recommended users in a recommendation list



(a) Precision (b) Accuracy

Figure 3.6: Precision and Accuracy

k	Precision-at-k
1	0.14315353
2	0.16457286
3	0.18903036
4	0.19455253
5	0.20481928
6	0.21046443
7	0.21572796
8	0.22233202
9	0.22504622
10	0.23011994

Table 3.4: Followee Recommendations Precision-at-k

Mean average precision (MAP@10)	0.199981913
Overall accuracy	0.988854305118

Table 3.5: Followee Recommendations Precision and Accuracy

This way, we managed to rank all users from the dataset according to the user profile and computed precision-at-k and mean average precision (MAP@10). Precision-at-k corresponds to the precision (information retrieval performance measure) calculated in the first k recommendations in the recommendation list. As relevant elements, we consider the user's ground truth followees made by the user (e.g., all ground truth followees in the recommendation list are considered true positives). We conducted experiments from k = 1 to 10. The results are shown in Tables 3.4 and 3.5.

As we can see, our method reaches a mean average precision score of 19.99%, indicating that our recommender can efficiently recommend followees. However, due to Twitter API Rate limits, our dataset is limited to 100 users, while the ground truth dataset contains all true followees. Therefore, we expect that precision-at-k would be higher if the data were complete.

3.10.3 Experiment with a large dataset

In the previous section, we presented the evaluation results of our method using a 100-user dataset. We used this dataset because the previous experiment aimed to compare our approach to the state-of-the-art, concerned the top-100 active users in one country. A larger dataset would make it impossible to test any collaborative filtering method due to Twitter's API rate limits regarding followee data. This small dataset raised questions regarding the efficiency of our method with less biased and more real-life datasets. Furthermore, we did not have a measure regarding the runtime of the proposed method. This section answers these questions by evaluating our tweet recommender using a large and not biased dataset. The evaluation of our followee recommender would require a large amount of user graph data (followers and followees), which we could not collect due to Twitter's API rate limits. Finally, we modified the KG by removing some nodes that are constantly miss-assigned by AlchemyAPI. For example, every tweet that the API could not assign to any other category was finally miss-assigned to the category "social network", because of the metadata or URL of the tweet.

3.10.3.1 Dataset

Our dataset consists of all public tweets (1% sample of all tweets) posted from 24 March until 23 April 2012, which were collected using the Public Streaming Twitter API. We requested only stream tweets written in the English language. During these 31 days of tweet crawling, we managed to store in compressed JSON text files 146 887 375 tweets (48,96 GB on disk). However, our dataset contained duplicate tweets and tweets without text or URL content, thus tweets that AlchemyAPI Taxonomy could not analyze. For this reason, we added a short preprocessing stage to remove duplicate and text-empty tweets from the dataset. To accomplish both preprocessing and further data manipulation needs, we imported the data into a PostgreSQL database.

3.10.3.2 Forward chaining validation and data integration

To test the efficiency of our tweet recommender, we choose as ground truth test dataset the data retrieved from users' retweets because we assume that when a user retweets a post, he is most likely interested in it. Moreover, since our recommender is meant to provide real-time recommendations based on past data (older user tweets), cross-validation could be problematic (e.g., interest changing, emerging events, new ToIs, etc.). The forward-chaining evaluation method can model the situation at the prediction time. Following this method, we divided the dataset into five subsets (four sets of 6 days tweets each and one of the seven-day tweets), respecting the chronological order of their publication to form our train datasets. Thus, our training sets are:

- TrainSet [1]: tweets and retweets from day1–day6
- TrainSet [2]: tweets and retweets from day7–day12
- TrainSet [3]: tweets and retweets from day13-day18
- TrainSet [4]: tweets and retweets from day19–day24
- TrainSet [5]: tweets and retweets from day25–day31

According to the forward-chaining evaluation, each test set should consist of the retweets from the directly consecutive set. Thus, our test sets are:

- TestSet [1]: retweets from day1–day6
- TestSet [2]: retweets from day7–day12
- TestSet [3]: retweets from day13–day18
- TestSet [4]: retweets from day19–day24
- TestSet [5]: retweets from day25–day31

The fourfold forward-chaining evaluation consists of the following steps:

- Fold 1: TrainSet [1], TestSet [2]

- Fold 2: TrainSet [1, 2], TestSet [3]
- Fold 3: TrainSet [1, 2 and 3], TestSet [4]
- Fold 4: TrainSet [1, 2, 3 and 4], TestSet [5]

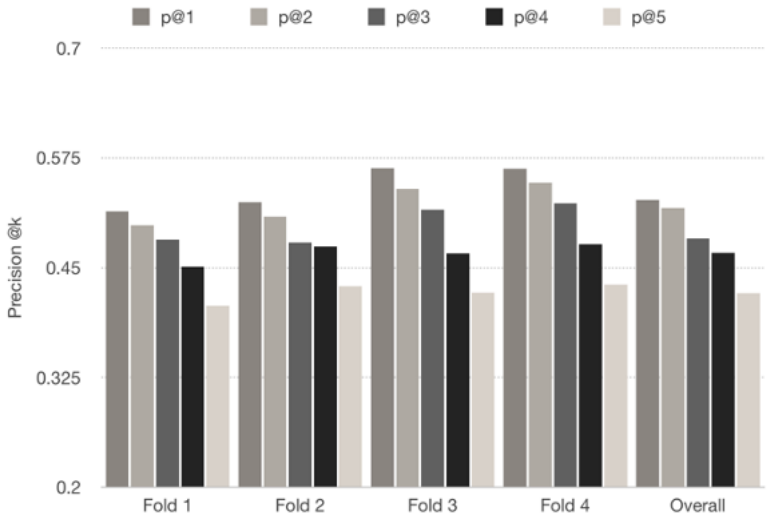


Figure 3.7: Forward-chaining Precision-at-k

For each training set, we constructed a new database table. These tables contain the streaming data in a reorganized way. Rows represent users who were active during the set's time period. Each row contains a user-id, a list of the tweets he posted during this time. Our Tweet Representation unit adds an extra column containing the user interest profile (sorted ToI list) based on the tweets he published during that period. We used only users who have posted three or more tweets (and/or retweets) during this month for our experiment. Finally, our train sets contain 200 039 unique users and their interest profiles. Each test set is a separate table where each line represents a retweet posted in chronological order because we wanted to simulate the streaming tweets entering our recommender. Each line contains the tweet-id, the user-id that posted this retweet, and a ToI assigned by the Tweet Representation module.

The results of our experiment from k = 1 to 5 are shown in Figure 3.7. Finally, we conducted an overall evaluation, where the training set contains all the tweets (no retweets) of the users, and the test set contains all the retweets of the users.

As we can observe, our method reaches an average precision score of 48,4% while the overall accuracy is 98.9%. This means that our recommender can successfully retrieve the interesting (retweeted) and not interesting tweets (true positive and true negative results) but still recommends some tweets that were not retweeted by the user (false positives). This could be caused, besides our recommender's weaknesses, since all users see not all retweets from the test set because they are not following every other user in the network. Hence, they cannot have retweeted something they could not have seen in their timeline, even if they are interested in it.

We can also observe that our method shows even better results in terms of efficiency when tested on a larger dataset. This could happen because the new dataset is

more consistent regarding time since the tweets and retweets were published within one month. Furthermore, it concerns many users increasing the possibility of having users that are followers-followees with each other in it (the possibility of retweeting content from one's followees is higher than from the random stream). Finally, the preprocessing stage that we described in Section 3.10.3.1 along with the modification in the KG has enabled the method to provide even better recommendations.

3.10.3.3 Runtime testing

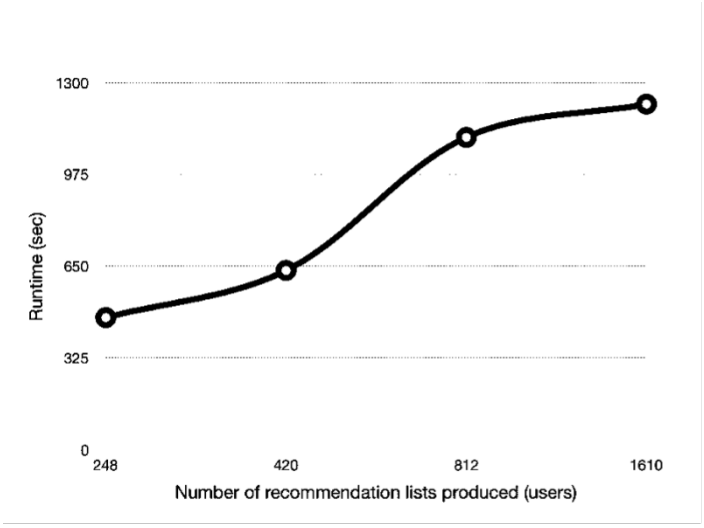


Figure 3.8: Recommender Runtime

Time scalability is essential in recommender systems, especially if they are designed for real-time use. Therefore, we tested our method for the first 1610 users to estimate our method's runtime scalability. Specifically, we measured the time needed for user profiling, tweet filtering, and top-5 ranking for 0 to 1610 users. As shown in Figure 3.8, our method can provide quick recommendations, revealing an opportunity to develop an online real-time tweet recommender application.

3.11 Conclusion

In this chapter, we presented a content-based method for personalized tweet and followee recommendations. This method is based on conceptual relations between users' topics of interest (ToIs). The method takes advantage of the objective relation between the ToIs of a user and a Knowledge Graph. We have shown that the recommendations based on these relations can reduce the effects of over-recommendation and over-specialization problems and can be used to capture the dynamic change of these interests too in a scalable way. The efficiency of the proposed method outperforms in many cases the previous state-of-the-art works. It exhibits even better results in terms of precision and time scalability when tested on a larger dataset.

Chapter 4

Automatic Generation of Feature-Agnostic Datasets for Fake News Detection in Social Media

4.1 Introduction and challenges

Over the last years, social media have become a primary channel of news information as more and more people tend to search and consume news from online platforms. Contrary to traditional news organizations, social media platforms allow millions of users to produce and access a vast source of information freely. However, the problem of fake news has grown into one of the most crucial issues for social media platforms, users, and news organizations. Although misinformation and disinformation phenomena exist since the birth of the printed press, new online platforms accelerate and boost their diffusion, posing new problems and challenges. Open and free access to social media has been used to cover fraud, misinformation, and manipulation mechanisms, that cause a severe negative impact on individuals and society. In this regard, algorithmic methods to automatically detect misleading content on social media are crucial to help mitigate those negative effects. Specifically, fake news detection can be treated as a news classification problem, and, therefore, machine learning algorithms can be used to develop detection models. Such models predict news veracity by relating training data instances to their already known class (true or fake). Hence, machine learning approaches are data-driven, meaning that the efficiency of the produced models depends on the training dataset size and quality.

These training datasets include features that capture various aspects of the data: news source trustworthiness, text writing styles, sentiment and semantic content of posts, social network structure, and propagation. Most works in the literature usually focus on a subset of properties: text, sentiment, stance, syntactic, style, headline, network structure, network propagation, semantic content, etc. Feature categories such as semantic and network propagation features are underestimated, and only a few available training datasets exist.

On another front, training datasets require sets of labeled news articles. Knowledge graphs can be very useful for providing training data for fake news detec-

tion techniques [98, 141, 129], since we accept that relations in a knowledge graph represent actual events. Such knowledge graphs can be constructed by leveraging data provided by fact-checking organizations that analyze and crosscheck news articles. Hence, the automatic generation of training datasets can rely on such external knowledge sources to acquire fake news content. The ability to filter such content leveraging the semantic web can contribute to automatically generate, curate and update large knowledge graphs (or knowledge bases) while minimizing inaccuracies and false information. This would greatly facilitate the early tracking of false claims within news stories.

However, there are certain characteristics of this problem that make it uniquely challenging:

- The manual process of labeling content and constructing training datasets is a time-consuming process that results in partial and outdated datasets referencing the more-or-less distant past. For example, such datasets are often missing some propagation points that refer to fake news posts that have been removed in the meantime by social media users and news websites. Moreover, outdated datasets cannot reveal current fake news spreading mechanisms because they tend to adapt to existing detection techniques. A useful training dataset needs to balance between accuracy (manual vs. automatic generation) and completeness (up-to-date and fast generation) of fake news diffusion data.

- There is significant diversity among existing datasets that raises severe problems to the reliable assessment and comparison of the detection models. User engagements with fake news in social media produce big, incomplete, unstructured, and noisy data. For instance, existing datasets suffer from significant limitations: incomplete news texts, few news sources, partial social network data (due to social platform API limits), a small number of news and social posts, and more.

- Most approaches focus on specific subsets of features that can be measured in multiple ways. Due to this, and to the best of our knowledge, no publicly available dataset exists that provides the necessary information to extract all state-of-the-art features for detection (text, user relationships, network analysis, news propagation, spatiotemporal information).

- The nature of fake news is not uniform [136]. Misinformation refers to fake or inaccurate information unintentionally spread, while disinformation concerns information intentionally false and deliberately spread. Figure 1 classifies misleading content according to facticity, which refers to the truthiness of the facts mentioned, and intention to deceive, which refers to whether the creator of the content intends to mislead. Misinformation includes parody, mistaken reports, and satire, all of which are low in their intention to deceive. Disinformation includes fabricated news and propaganda, which are high in their intention to deceive. Propaganda is different from fabricated news in that it contains true facts within an extremely biased context. Intention detection methods (horizontal dotted line in Figure 1) can be used to distinguish between disinformation and misinformation, while fact-checking methods (vertical dotted line in Figure 1) can be used to estimate the truthiness of facts. In this chap-

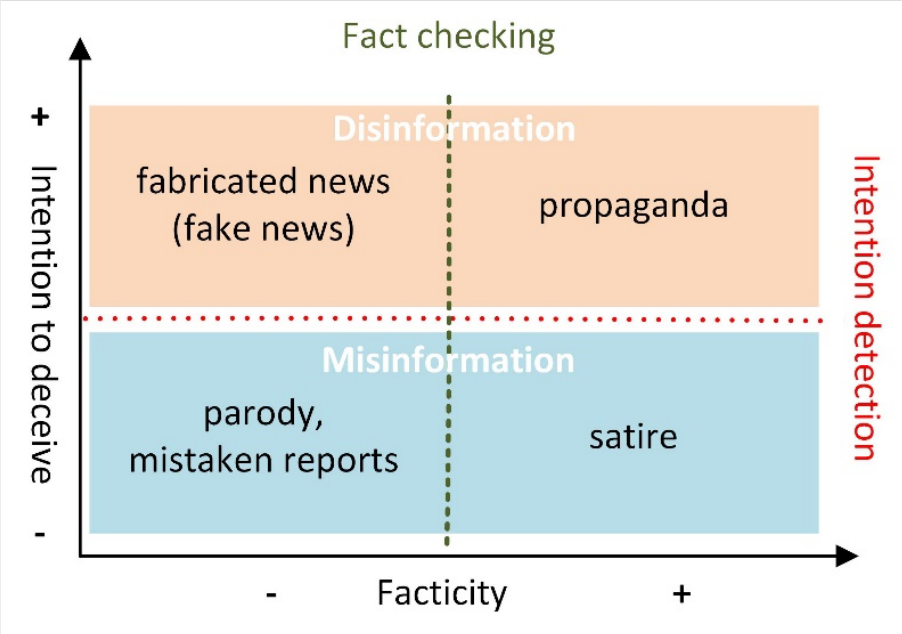


Figure 4.1: Misinformation Typology

ter, the term fake news denotes inaccurate information, both misinformation, and disinformation.

- Fake news is usually related to newly emerging, time-critical events, which may not yet have been properly verified by existing knowledge bases and fact-checkers due to the lack of corroborating evidence or claims. Therefore, the network-based detection approaches that rely on fake news propagation patterns become increasingly important since they do not require early veracity knowledge. As detecting such content early can help prevent further propagation on social media, it is crucial that training datasets contain significant fractions of propagation data using efficient propagation representations, which can be later used as features to train machine learning classification models. Notice that social network diffusion features have not received as much attention so far.

As a solution, we propose PHONY, an infrastructure that automates the generation of *feature-agnostic* datasets. Feature-agnostic datasets are not constructed with a rigid set of features in mind but allow the extraction of ad hoc datasets using feature-specific scripts according to the machine learning approaches used. Although the complete automation of the dataset generation process is not yet possible, it is essential to automate the dataset generation process as far as possible. Such automation will result in the generation of uniform, flexible, and up-to-date silver standard[1] datasets and the reliable assessment and comparison of fake news detection methods.

PHONY users can choose the fact-checking source, time period, and fake news type. The resulting datasets contain fake news and their propagation footprints

[1] A silver standard corpus is usually defined as a noisy set of ground truth annotations provided automatically by state-of-the-art algorithms, while gold labels are higher quality annotations created by expert annotators [90].

on Twitter. The central idea behind our approach is described in [102]. PHONY makes use of the fake news stories provided by curated fact-checking websites and produces datasets that include social media posts that refer to those fake news stories. Specifically, PHONY is based on building an incremental tweet inverted index comprising streaming tweets. This index consists of broadened tweets that comprise the tweet text and metadata as well as web data related to each tweet. The generation of a new dataset starts with collecting fake news items from fact-checking websites and then analysing and using them as queries to the inverted index.

PHONY generates feature-agnostic datasets, hence datasets that do not contain metrics of specific classification features. This allows users to freely choose the features and measurements that better suit their classification and detection methods. In fact, all features encountered in the literature can be directly extracted from our dataset by simple scripts and algorithms since PHONY provides text, user, network, and propagation data. PHONY datasets include time information, which means that each feature can be treated as a time-series. PHONY provides uniform and updatable data that can significantly contribute to developing effective fake news detection models since fake news creators evolve and adapt to avoid the current detection methods.

To showcase PHONY, we present in detail a new dataset created by the proposed infrastructure. Since we are interested in the propagation of Greek fake news, we created a large-scale feature-agnostic dataset for fake news detection by utilizing Greek tweets that refer to fake news content. The Greek PHONY Dataset[2] is available to the research community. In the context of another ongoing work, we continuously update this repository, expand it with new fake news sources and the Twitter footprints of new and emerging fake news, and maintain its completeness. A sampling evaluation showed that the PHONY dataset reached an average precision of 77,5%. We also conducted an exploratory data analysis on a snapshot of this dataset and a brief statistical overview of the corpus to validate the dataset's quality and reveal some interesting diffusion patterns and topics of interest relevant to the Greek Twittersphere.

4.2 Overview of fake news features

Fake news detection methods in social media are based on models trained on features that have been extracted from social media datasets. Thus, the efficiency of the produced models relies highly on the quality of the training datasets.

4.2.1 Frequently Used Detection Features

Most misleading content detection methods utilize linguistic features by applying NLP methods on text extracted from social media posts, enriched with sentiment, topic, unigram detection approaches [147, 88, 27, 58, 78, 166, 155, 57, 168, 106] or by using domain specific entities such as hashtags, URLs, emoticons and quoted text [88, 27, 58, 166, 106]. Misleading intention has also been identified by using features that capture deceptive indications in writing styles (e.g. lying [10]), hyper-partisan

[2]The dataset is available for users at https://imisathena-my.sharepoint.com/:f:/g/personal/danae_imis_athena-innovation_gr/EjODLOshRWVNvj-VGcJ-IwYBZUv_r3DHnw9FdDOzGmiDSw?e=en00ya

language [104] and eye-catching headlines (i.e., clickbait) [30]. Furthermore, recent works extracted such features from the comments and replies to a news post, assuming that misleading news could raise skeptical reactions by readers. These features are calculated for each post or for a group of relevant posts and aim to capture the response volume, stance (i.e. opinion) [66], topic [88], and credibility [27, 66]. Fewer methods try to capture the temporal variations of these features, including unsupervised embedding methods, such as recurrent neural networks (RNN) [116, 87], time windows, and mathematical features such as SpikeM [57] that capture the shape of the time series for these metrics. Recently, advanced NLP models were applied to spot deception intention utilizing deep syntax (rules that describe the syntax structure) [49] and rhetorical structure (to capture the differences between deceptive and truthful sentences) [115]. Deep network models, such as convolutional neural networks (CNN), have also been applied to classify news veracity [151].

4.2.2 Semantic Features

The complexity of language, humorous and satire texts, and the background context of news stories pose certain challenges in this problem. Therefore, the comparison of news stories to some existing knowledge requires capturing the reasoning behind news stories, hence inferring the semantics of news texts. In this direction, various semantic features have been proposed in previous works. First, semantic proximity or semantic relatedness [35, 36] is based on various shortest path calculations on the knowledge graph. The intuition behind these approaches is that a claim, represented as an (S, P, O) triple, is true if it exists as an edge on the knowledge graph or a short path linking its subject to its object within the knowledge graph. Semantic relatedness has also been used to identify satirical and humorous content by comparing semantically different parts (lead and final sentence) of news articles [114].

Moreover, semantic relatedness can be calculated using word taxonomies such as Wordnet [114] and relatedness measures such as word-to-word semantic similarity score. Some works claim that different paths between two facts provide different evidence for a claim, even if they have the same length. Therefore, they claim [35, 129] that the definition of path length used for fact-checking should account for more information-theoretic parameters such as specificity and generality.

Another approach for semantic fake news detection is the representation of news articles as word embeddings to capture relational similarities and be used as input to neural network classifiers [23, 161]. In [129], authors calculate the average vector for the title and separately for the content of the news article and compute the similarity between them, based on the assumption that a low similarity is highly indicative of click-bait articles, whose content differs from what the content of the title. Finally, embedding representation has been applied on knowledge graphs for computing semantic similarities [98].

4.2.3 Network Features

From another perspective, misleading news detection can be approached under the light of network diffusion. For example, such content is likely to be posted by social bots (automatic behavior), thus, user-based features [88, 27, 78, 162] provide useful

information for misleading news detection [89, 88, 27, 78, 57, 155, 124]. Moreover, identifying misleading sources in a social network allows early containment of the epidemic-like spreading and a crucial indicator for analyzing news's truthfulness. However, since propagation paths are often unknown, source detection task has attracted a lot of interest [15, 65, 159, 123, 121, 122, 47, 42, 48, 153, 170, 93]. A significant aspect of the problem is the analysis of the diffusion network that is formed by the spread of misleading news content [78, 66, 116]. Therefore, diffusion analysis can be used to extract patterns and signals that correspond to misleading content spread. Recent works claim that diffusion of such content demonstrates unique temporal patterns [78]. However, the propagation is affected by the existence of strong user communities (based on interests, topics, relations, etc.). Users are selectively exposed to certain kinds of news since their timeline consists of their friends' posts or by like-minded followees. Therefore, they interact with news that likely promotes their favored opinions [107], resulting in groups with like-minded people. This echo-chamber effect enhances the propagation of misleading content because users consider their like-minded sources as credible and because they tend to believe information with which they interact more often [146].

4.3 Fake news diffusion: organising features for machine learning models

In the context of this work, we explored the range of the misinformation and disinformation detection features encountered in the literature. We analyzed, compared, linked, and joined all 256 features that are comprised in the previous works described in section 4.2. The following steps sum the approach that we followed and correspond to each column of the table presented in detail in appendix A:

1. **Feature cluster**: we merged the 256 features in 80 clusters of similar variations of the same feature

2. **Feature name**: the name of the feature as encountered in the literature

3. **Type**: we grouped them based on the into five types: Content, Network Structure, Topic Propagation, User Meta, Tweet Text

4. **Type of analysis**: we grouped them based on the type of specific analysis they require into sixteen categories: Community, Flow, Friendship Reciprocity, Grammar/ Syntax, NLP, Influence, Potential Impact, Profile, Role, Search, Text Sentiment Analysis, Textual Patterns, Topic Analysis, Tweet Meta, Tweet Volume, User Volume

5. **Measurement**: type of measurement

6. **Granularity**: granularity of measurement

7. **Time**: we classified them based on whether they can capture the time aspect (time-series)

8. **Definition**: we offer the exact definitions if available

Observing the results of this literature review revealed the multitude of features used to study different aspects of propagation, measured in different ways and different granularity. This diversity leads each bibliographic work to use different training datasets, making the comparison of fake news detection models almost impossible.

Moreover, this complete feature typology can contribute to developing new methods for detecting misinformation and disinformation based on the content (factual, narrative) and the diffusion process of news in social media. Specifically, it can help develop machine learning approaches in which different sets of features extracted from the ground truth datasets will be used to detect the various forms of misleading content. Instead of building a single model for all the aspects or types of misleading content, this typology can facilitate researchers to blend the features and individual models by leveraging ensemble learning techniques.

4.4 Fact-Checking of News Stories

The generation of solid ground truth fake news datasets requires labeled fake news instances from trusted curators. Therefore, training datasets should rely on knowledge-based approaches, which are called fact-checking. These approaches cross-check claims against true knowledge in order to assess news veracity. Most existing methods rely on human domain experts (PolitiFact [3], Snopes [4]) or crowdsourcing (Fiskkit [5]). However, these approaches are intellectually demanding and time-consuming, limiting the potential for high efficiency, scalability, and early detection.

Automatic fact-checking relies on techniques borrowed from information retrieval, semantic web, and linked open data (LOD) research. They are based either on knowledge graphs (DBpedia [6], YAGO [7], NELL [8], Knowledge Vault [9], etc.) or on knowledge derived from open-web data and can provide a scalable veracity classifier. Knowledge graph approaches check whether the claims in posts content can be inferred from existing facts in the knowledge graph (existing links or strong reachable connections) [126, 35], and tackle the problem as a link prediction problem [126] or as the shortest path finding problem [35]. Moreover, open-web approaches [103, 157, 89] compare posts with open-web claims in terms of consistency and frequency (stance and credibility assessment) or use fact extraction methods to form new knowledge bases. Fact extraction methods aim to extract structured knowledge (e.g., Subject, Predicate, Object triples) from the unstructured information found on the open web using knowledge extraction models and tools [98] along with entity recognition and advanced NLP techniques.

[3] https://www.politifact.com/
[4] https://www.snopes.com/
[5] https://www.fiskkit.com/
[6] https://wiki.dbpedia.org
[7] https://github.com/yago-naga/yago3
[8] http://rtw.ml.cmu.edu/rtw/kbbrowser/
[9] https://developers.google.com/knowledge-graph

4.5 Existing Datasets

Though the research community has produced several datasets for fake news detection, no automatic system exists for creating fake news datasets in social media to the best of our knowledge. The most significant state-of-the-art datasets are listed below.

BuzzFace: This dataset [117] contains 2282 news stories that were published during September 2016 on nine Facebook pages that correspond to nine news outlets. These articles have been annotated by BuzzFeed, a famous fact-checking organization, into four categories: mostly true, mostly false, a mixture of true and false, and no factual content. BuzzFace contains the corresponding comments and reactions, article texts, images, links, and embedded tweets, reaching over 1.6 million text items.

PHEME: This dataset [171] contains rumors associated with nine different news stories and Twitter conversations initiated by a rumor-related tweet. The authors used Twitter's streaming API to collect tweets from breaking news and specific rumors that are identified a priori. Collection through the streaming API was launched straight after a journalist identified a newsworthy event likely to give rise to rumors. The collected tweets have been annotated for support, certainty, and evidentiality. The sampled dataset includes 330 rumorous conversational threads.

USPresidentialElection: The authors [128] retrieved tweets referring to 57 rumors from October 2011 to December 2012. These tweets were gathered by matching specific keywords for each of these 57 rumors. After a tweet preprocessing stage, the authors assembled a set of keywords to identify the rumor-related tweets based on the description of each rumor offered by rumor debunking sites. These keywords were then combined through logical expressions, forming queries manually repeatedly tested by two authors. Finally, human coders measured each tweet for two variables: whether the tweet was actually about the rumor and its author's attitude (endorsing, rejecting, or unclear).

FakeNewsNet: This repository [130] contains two datasets with news content, social context, and spatiotemporal information. To collect ground truth labels for fake news, the authors utilized two fact-checking websites to obtain news contents for fake news and true news. User engagements related to news articles were collected using Twitter's Advanced Search API. The search queries were formed from news articles' headlines, and the obtained tweets were enriched by replies, likes, reposts, and user social network information.

CREDBANK: This dataset [91] comprises more than 60 million tweets grouped into 1049 real-world events, each annotated for credibility by 30 human annotators. These tweets were collected from all streaming tweets over three months, computationally summarized into events, and finally routed to crowd workers (Amazon Mechanical Turk) for credibility assessment.

SomeLikeItHoax: This dataset [134] consists of all the public posts and their likes from a list of selected Facebook pages during the second semester of 2016. These Facebook pages were divided into two categories: scientific news sources vs. conspiracy news sources. The authors assumed all posts from scientific pages to be reliable and all posts from conspiracy pages to be "hoaxes". The dataset contains 15,500 posts from 32 pages (14 conspiracy and 18 scientific) with more than 2,300,000 likes.

TraceMiner: The authors [156] retrieved tweets (using Twitter search API) related to fake news by compiling queries with a fact-checking website (Snopes). The dataset contains 3600 messages from which 50 percent are referring to fake news.

Vosoughi et al.: To construct this dataset [148] the authors, firstly, scraped fact-checking websites (https://www.snopes.com/, https://www.politifact.com/, https://www.factcheck.org/, https://www.truthorfiction.com/ and https://hoax-slayer.com/), collected the archived rumors and parsed the title, body and verdict of each rumor. The rumors were then divided based on their topic (Politics, Urban Legends, Business, Science and Technology, Terrorism and War, Entertainment, and Natural Disasters) using LDA and human annotators. Afterward, they accessed the complete Twitter historical archives to extract all reply tweets containing links to any of these rumors through the fact-checking websites. Then, unrelated tweets were filtered out through a combination of automatic and manual measures. For each reply tweet, they extracted the original tweet they were replying to and then extracted all the original tweet's retweets. Each of these retweet cascades is a rumor propagating on Twitter. The dataset contains 126285 rumor cascades corresponding to 2448 rumors.

4.5.1 Comparison

Examining the background work, one concludes that existing approaches suffer from various limitations resulting in datasets that do not provide all possible features of interest. For example, some datasets have been produced based on a small number of news outlets (BuzzFace), include a small number of rumors (USPresidentialElection), or have a very restricted definition of real and fake news (SomeLikeItHoax). Moreover, some datasets cover a short period (BuzzFace), most of them are outdated or cover a limited topical range (USPresidentialElection). Another disadvantage within the scope of developing automatic dataset creation systems is that many state-of-the-art datasets rely heavily on human annotators (CREDBANK, USPresidentialElection). In addition, some datasets suffer from a very narrow definition of news propagation (BuzzFace, CREDBANK). For example, BuzzFace uses the Facebook Graph API to collect data on "reactions" to the original articles. Still, independent posts that spread this news are missing, as is news propagation's temporal information.

Furthermore, some approaches underestimate or omit social bots' role in the news propagation process by removing them from their corpus (CREDBANK, Vosoughi, et al.). The most crucial limitation that almost all datasets suffer from is the social

Table 4.1: Existing Dataset Comparison

Dataset Features	Diff. Fake News Types	Automated Generation	Full Articles	Social Posts	Network
BuzzFace			✓	✓	
PHEME				✓	
USPresidentialElection				✓	✓
FakeNewsNet			✓	✓	✓
CREDBANK				✓	
SomeLikeItHoax				✓	
TraceMiner			✓	✓	✓
Vosoughi et al.			✓	✓	✓
PHONY	✓	✓	✓	✓	✓

network API limits (Facebook, Twitter) that give partial access to social media information. One dataset (Vosoughi et al.) claims to have access to the full Twitter History, but this dataset is not available to the research community.

Table 4.1 depicts a comparison between these datasets and PHONY, the dataset we created for Greek tweets using the process described in this chapter.

4.6 Objectives of our Approach

The aim of the PHONY Infrastructure is the automation of the generation of feature-agnostic datasets. These datasets will allow users to generate ad hoc silver standard datasets using simple feature-specific scripts and algorithms. However, this task has particular requirements that need to be considered when designing and implementing the system.

- **Size and quality of the generated datasets:** The system should generate datasets that cover a wide type variety and a large number of fake news, as well as provide complete representations of their propagation on the Twitter network. Moreover, due to the limited size of tweet messages, likely, fake news will not spread directly through the tweet text. Instead, fake news spread via links to external news websites. However, websites containing fake news often disappear after a short period, and the social traces of news items are often lost. Hence, it is crucial that datasets cover the full life cycle of such news.

- **Automatization and standardization of dataset generation:** Although the full automation of the dataset generation process is not yet possible, automating the dataset generation process as much as possible is essential for the reliable assessment and comparison of fake news detection methods. Moreover, the selection of features and the way of calculating those features should not restrict researchers when selecting available datasets. Our goal is to generate feature-agnostic datasets, hence datasets that can be used as a basis for creating ad hoc feature datasets. Moreover, it is crucial to generate feature-agnostic datasets such that all features encountered in the literature can be directly extracted using specific scripts and algorithms. Let alone that the analysis of such feature-agnostic datasets could facilitate researchers to derive new features for detecting fake news in social media.

- **Dynamically updatable datasets:** It is observed that fake news creators and their spreading mechanisms evolve and adapt to avoid the current detection methods. Hence, the delayed analysis of such datasets might not be useful for detecting the propagation of emerging fake news stories. Therefore, it is of great importance that the produced datasets can be continuously and incrementally updated.

4.7 PHONY Infrastructure

PHONY is an essential part of our ongoing research for developing models to detect fake news content in streaming data automatically. The datasets that are created automatically will be used to extract specific features and train machine learning classifiers.

4.7.1 Overall System Architecture

Figure 2 depicts the overall architecture of our automatic fake news dataset generation system. PHONY continuously collects the streaming tweets using the Twitter Streaming API and incrementally building an inverted tweet index. When a new dataset needs to be generated, the system collects fake news items from fact-checking websites and uses them to query the inverted index.

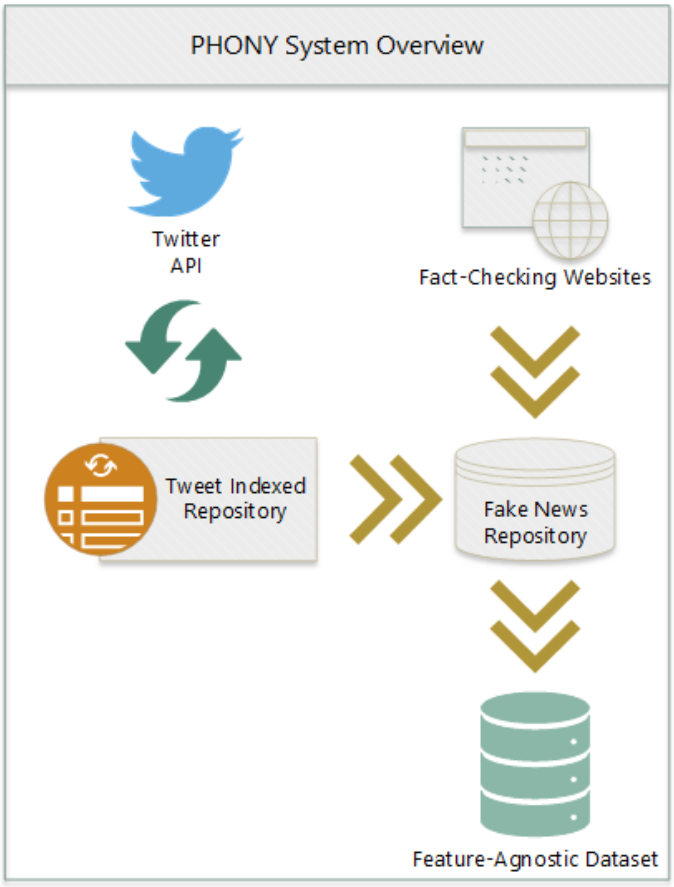


Figure 4.2: PHONY Infrastructure Overview

PHONY consists of three asynchronous units: The Tweet Index Creation unit, the Fake News Collection and Analysis unit, and the Dataset Generation unit. The Tweet Index Creation unit creates an inverted index of streaming tweets incrementally. This unit is running continuously regardless of whether there is an ongoing fake news generation process. Fake News Collection and Analysis unit is activated on demand. It is responsible for collecting and analyzing fake news stories based on three criteria chosen by the user: (a) fake news source (fact-checking website), (b) time period, and (c) fake news type (fabricated news, propaganda, satire, parody, etc.). The Dataset Generation unit is then activated through a query process and generates the fake news stories' Twitter diffusion footprints. Specifically, queries are formulated based on the analysis performed in the Fake News Collection and Analysis unit and are then run on the index generated by the Tweet Index Creation unit. These three main units are described in detail in the following sections.

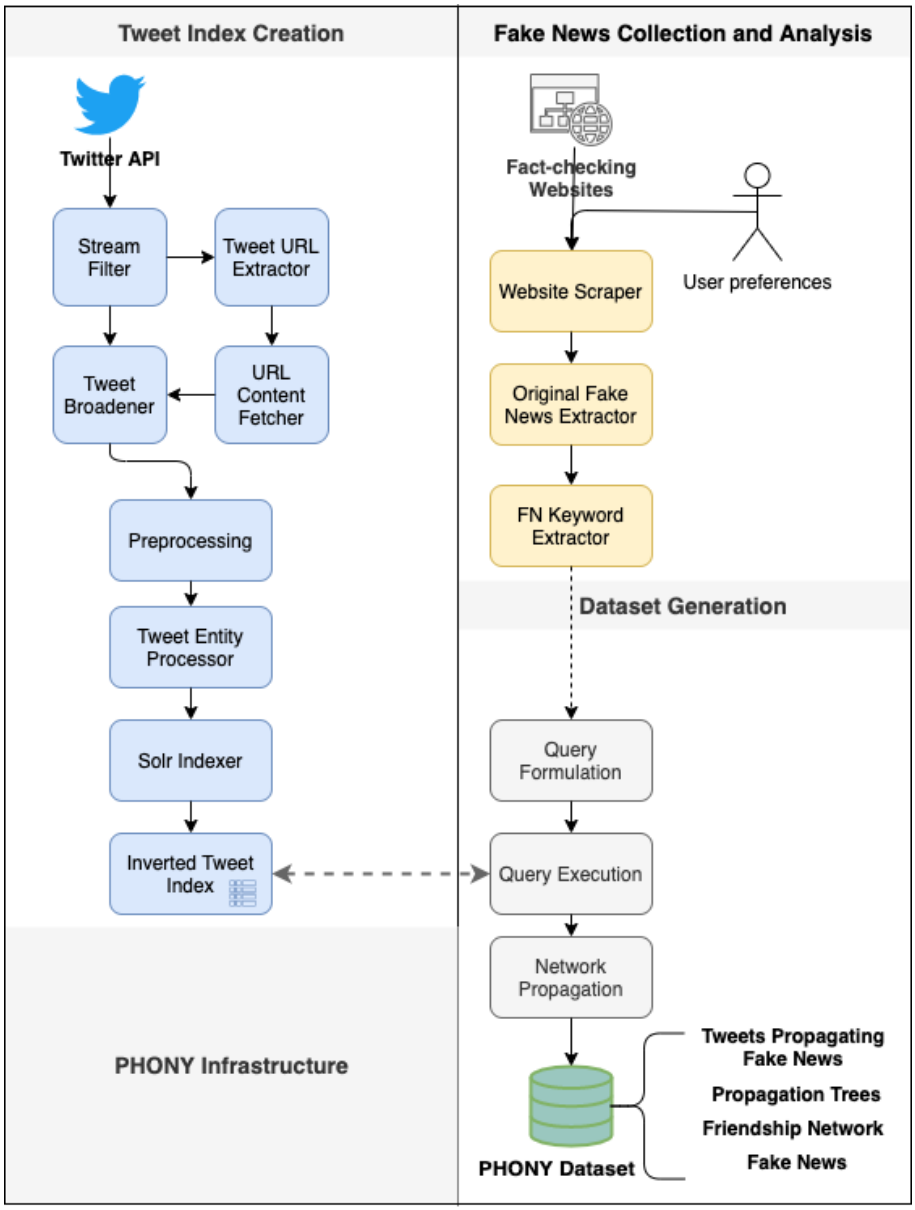


Figure 4.3: PHONY System Architecture

4.7.2 Tweet Index Creation

To dig out fake news stories scattered throughout the social network, we need an efficient way to organize the streaming tweets. Our system operates by extending the Twitter Streaming API responses, creating a large set of JSON files. JSON objects offer an intuitive and descriptive way to represent our data and are also used by the Twitter Streaming API as response values. However, directly querying the JSON files would mean that the system would scan every document in the corpus, requiring considerable time and computing power. To address this problem, our method creates an inverted index of all the posts that have been previously filtered from the stream and subsequently broadened using the main content of the included URLs. Therefore, in this step we describe the creation of an inverted tweet index, a hashmap-like data structure that directs from a word to a tweet. The Tweet Index Creation process is running continuously, processing every single streaming tweet regardless of the fake news dataset creation process. This way, a possible

tweet deletion will not cause the propagation data loss to the generated datasets. As a result, the inverted index is incrementally rebuilt on a regular basis. The basic modules included in this unit are the following.

Stream Filter: The most crucial limitation that almost all existing datasets suffer from are the social network API limits (Facebook, Twitter) that give partial access to social media information. Specifically, Twitter maintains two publicly accessible APIs: The Search API and the Streaming API. The Search API returns fewer results than the Streaming API, which emphasizes central users and more clustered regions of the network, while peripheral users are less accurately represented or absent. In addition, Search API results are not a random sample of overall Twitter activity (most popular tweets are promoted). Therefore, a simple search of fake news keywords through the Search API would not lead to the desired outcome: the returned tweets would be few and sampled in a biased way. Therefore, our system uses the Streaming API. However, this API samples a random 1 percent of the whole Twitter activity, leading to incomplete and diverse results. Despite that, the fact that we are interested in the propagation of Greek fake news allows us to disregard this limitation to some extent. Specifically, our system uses the Streaming API by filtering the global Twitter feed based on a set of common Greek words.

Given that tweets written in Greek do not reach the 1 percent of the global activity, this method is adequate to ensure the collected data's completeness making Greek fake news diffusion in Twitter a good case study. This module filters the collected streaming tweets using a set of language-dependent keywords, adequate to ensure the collected data's completeness. These keywords are then used as parameters of the Twitter Streaming API services. The module stores the collected data in compressed JSON Lines files on an hourly basis.

Tweet URL Extractor: This module extracts the URLs included in the tweets that the Stream Filter has previously stored. The extracted URLs are stored in compressed text files.

URL Content Fetcher: This module processes the URLs that have been extracted by the Tweet URL Extractor and retrieves the HTML content of each URL using HTTP requests. The HTML content for each URL is stored in a MongoDB database.

Tweet Broadener: This module processes the URLs that have been extracted by the Tweet URL Extractor and retrieves the HTML content of each URL using HTTP requests. The HTML content for each URL is stored in a MongoDB database

Preprocessing: In this module, the enriched text of each tweet goes through a preprocessing procedure. NLTK3 and a language-dependent stemmer were used for the natural language processing functions: tweet text is converted to lowercase, URLs are removed, the text is divided into words, punctuation is removed, common words (stop-words) are removed, and the stem of the remaining words is acquired. The resulting words are added to the tweet. Following, the processed tweets are stored in compressed JSON Lines files.

Tweet Entity Processor: This module reads the tweet JSON object from the compressed files and converts it to the form that Solr requires while discarding unnecessary fields that do not need to be stored by Solr.

Solr Indexer: Specifically, this module generates the inverted index using the Apache Solr [17] search engine. The inverted index essentially links each term within the tweet collection to the broadened tweets containing the term. It is important to

note that the tweet collection contains all tweets that have been collected from the Twitter stream, without regard to their possible association with fake news.

4.7.3 Fake News Collection and Analysis

The Fake News Collection and Analysis unit runs on demand whenever we need to create a fake news dataset based on the user's three criteria: fake news source, time period, and fake news type. Specifically, this unit is responsible for collecting ground truth fake news stories from the web and analyzing their text content in order to extract representative keywords. Those keywords are used for querying the tweet index. The basic modules included in this unit are the following.

Website Scraper: This module retrieves a list of fake news from any fact-checking website the user chooses to use. However, these organizations do not provide API access to their databases or a coherent fake news dataset. In fact, many fact-checking organizations provide fake news data in the form of a simple web page or a blog. Therefore, to create a system that can collect fake news data from multiple fact-checking sources, we opted to use the scraping of such websites to obtain a list of fake news content along with the textual information that demonstrates their validation or disproof process. Moreover, our system stores fake news types by scraping the corresponding labels since fact-checkers provide such labels for each fake news item (fabricated news, propaganda, conspiracy theories, unscientific news, clickbait, scam, etc.). Hence, our system users can choose the fact-checking source and the type of fake news from a customized list and feed the module with relevant websites and blogs. This list is customized because the scraping process requires a customization level for each webpage and because fact-checkers do not follow a uniform tagging system for the aforementioned fake news types. Each article's title and text are retrieved and then stored in a MongoDB database using HTTP requests.

Original Fake News Extractor: Fact-checkers provide critical analysis and evaluation of the truth of each article. However, this content is not the original fake news story, and therefore, it cannot be directly used to trace fake news in social media. To find the original fake news, it is necessary to analyze these articles and retrieve the fake news articles from their original sources. This is achieved through a recursive process of detecting original sources of fake news from the fact-checkers' articles (extract external links). Then, using HTTP requests, the HTML content of each original URL is retrieved, and the main article content is extracted and stored in MongoDB. However, news web pages often contain advertisements, pop-up ads, images, and external links around an article's body. Hence, we can further optimize our web content scraping process to filter irrelevant content out completely. In the future, we aim to minimize human effort in the process of creating such datasets by developing fully automated scraping modules that will be able to scrape multiple non-uniform sources.

FN Keyword Extractor: At this point, we could define the diffusion of misinformation in social media as the propagation of those links. However, this approach would miss a large amount of fake news diffusion without reference to external sources via links and would not be able to retrieve fake news from sources that the fact-checkers have not discovered yet. For this reason, we apply content analysis to the original articles, and our system extracts a keyword vector for each original fake news story. In this module, the system extracts keywords for each fake news

article stored in the database. Specifically, keyword vectors are extracted using the TF/IDF method to acquire the most representative keywords for each fake news case.

4.7.4 Dataset Generation

This unit generates the Twitter diffusion footprints of the fake news stories by forming queries based on the fake news analysis and running them against the inverted index. Specifically, this unit constructs the propagation representation of the fake news stories in the Twitter network and the friendship network among Twitter users participating in the fake news diffusion process. The three modules included in this unit are the following.

Query Formulation: Our system uses the produced keyword vectors for each fake news to form queries performed on the inverted index of tweets. However, the query formulation requires a keyword selection strategy. We create queries by combining the top n terms of each fake news story with the AND and OR operators. We chose number n based on the average number of words per news story (a news story may consist of several articles). Specifically, we evaluated a sample of the collected fake news stories, and the results showed that

$$n = \ln(avg(length(fake\_news\_story))) \tag{4.1}$$

can offer us a balanced representation. This module forms the queries combining the keywords of each fake news instance with the AND operator. If the fake news story is found on more websites (multiple instances), our system uses the AND operator to combine the keywords of each instance and the OR operator to combine all instances' queries.

Query Execution: In this module, the system runs a query on the inverted index for each fake news story. The module returns a set of tweets from the index that satisfy the query. Hence, a set of tweets is assigned to each fake news story.

Network Propagation: This module constructs and stores the propagation representation of fake news on Twitter. A news story on Twitter starts with a user posting a claim about a topic or an event using text, multimedia, or links to outside sources. People then propagate the news by retweeting the post. Users can also comment and discuss the news by replying and mentioning other users to encourage a larger audience to engage with the news. Twitter APIs do not, however, provide the exact path of retweets since only the original tweet is provided when a retweet takes place. We, therefore, have to extract the diffusion network, which consists of one or more propagation graphs.

Based on the intuition that Twitter users retweet content that they have interacted with, we assume that a user retweets content that they have first seen on their timeline, hence, content that has been posted by one or more of their followees. In the case that a single followee has formerly posted the same tweet, we attribute the propagation of the tweet to the specific followee's post. In the case that multiple followees have formerly posted the same tweet, we have three choices as per the attribution strategy. We attribute the tweet's propagation to all the specific followees' posts, the most recent followee's post among the specific posts, or the most influential followee's post among the specific posts. Each of the three attribution

strategies leads to constructing a propagation graph for a particular sequence of tweets consisting of an original tweet and all the associated retweets.

Note that, in order to construct the propagation graphs for a news story, we do not need the complete follower/followee network for all the users participating in the propagation process. We only require the friendship relationships (the followees that a user has) for the users that participated in the propagation process by retweeting. To accomplish that, we perform targeted Twitter API calls to retrieve the list of friends (followees) for each user that posted a retweet. The module stores the diffusion network (propagation graphs) and the friendship network (user followees) for each fake news story in a MongoDB database.

4.8 A Concrete Example

As an example, consider the recent news story, which claims that cannabis can cure severe health conditions such as cancer[10]. In recent years, we have observed increasing claims that cannabis is an effective way of treating cancer. Occasionally, we see articles that make "aggressive" claims about the effectiveness of cannabis against cancer. According to a recent US FDA announcement, there is no evidence that cannabinoids can cure cancer, and companies that market-related products (in the US) should stop making such claims. It even highlights the risk of using these products because interactions with regular medicines, side effects, and the correct dosage are unknown. This particular claim showed up in the Greek news-sphere in July 2017, and we found it debunked on the Greek fact-checking website *Ellinika Hoaxes* [11]. In Greek PHONY Dataset, it is the Fake News Story with id=215.

In order to generate the propagation dataset for this particular fake news story, our system scrapes the specific article from *Ellinika Hoaxes* and stores the title ("Can cannabis cure cancer?") and the text of the article in a MongoDB database through Website Scraper. Then, Original Fake News Extractor analyzes this text to find the fake news stories' original sources. This recursive NLP process results in the following five fake news source links (instances):

1. "A cannabis protocol for the treatment of cancer": https://archive.fo/8KdYc
2. "How cannabis cures cancer": https://archive.fo/XpqEc
3. "Medical Cannabis: How It Cures Cancer, Children's Epilepsy, And Not Alone": https://archive.fo/vQ1ub
4. "Cannabis oil and cancer": https://archive.fo/YtWO2
5. "Ambrosia eliquids electronic cigarette replenishment fluid with CBD - Cannabis vapor has health benefits": https://archive.fo/W8yJZ

Then, using HTTP requests, the HTML content of each original URL is retrieved, and the main article content is extracted and stored in MongoDB. Next, FN Keyword Extractor analyses the original articles' content and extracts a representative keyword vector for each original fake news story using TF/IDF. The Query

[10] We translated the examples from Greek to English to be able to use them within the chapter.
[11] https://www.ellinikahoaxes.gr/2017/07/03/can-cannabis-cure-cancer/

Formulation unit calculates the number of terms in each query. In this case, n=5, based on eq. (1). For each fake news story, the system constructs a query using the AND operator to combine the top-5 keywords, as follows:

1. Query1: ”cannabinoid” AND ”cancer” AND ”cannab” AND ”marijuan” AND ”treatment”
2. Query2: ”cannabinoid” AND ”thc” AND ”cannab” AND ”cell” AND ”receptor”
3. Query3: ”cannab” AND ”hemp” AND ”simpson” AND ”rick” AND ”oil”
4. Query4: ”cannab” AND ”oleum” AND ”oil” AND ”plant” AND ”solvent”
5. Query5: ”ambrosia” AND ”cannab” AND ”cannabidiol” AND ”liquid” AND ”extract”

The final query is formed by combining the five instance queries into one using the OR operator:

Query:

```
(broadened_text:"cannabinoid"
AND broadened_text:"cancer"
AND broadened_text:"cannab"
AND broadened_text:"marijuan"
AND broadened_text:"treatment")
OR (broadened_text:"cannabinoid"
AND  broadened_text:"thc"
AND  broadened_text:"cannab"
AND  broadened_text:"cell"
AND  broadened_text:"receptor")
OR (broadened_text:"cannab"
AND  broadened_text:"oleum"
AND broadened_text:"oil"
AND broadened_text:"plant"
AND broadened_text:"solvent")
OR (broadened_text:"ambrosia"
AND  broadened_text:"cannab"
AND  broadened_text:"cannabidiol"
AND  broadened_text:"liquid"
AND  broadened_text:"extract")
```

Next, the Query Execution module runs the query on the Solr inverted index of broadened tweets and stores those that satisfy the query in the MongoDB. Broadened tweets are objects that consist of the original tweet fields (created_at, entities, extended_entities, favorite_count, favorited, filter_level, id, id_str, is_quote_status, lang, possibly_sensitive, retweet_count,retweeted, source, text, timestamp_ms, truncated, user) and an extra field (”broadened_text”), which wraps the texts of links to external websites contained in the tweet. For example, one of the 652 broadened tweets that are returned by the query is the following:

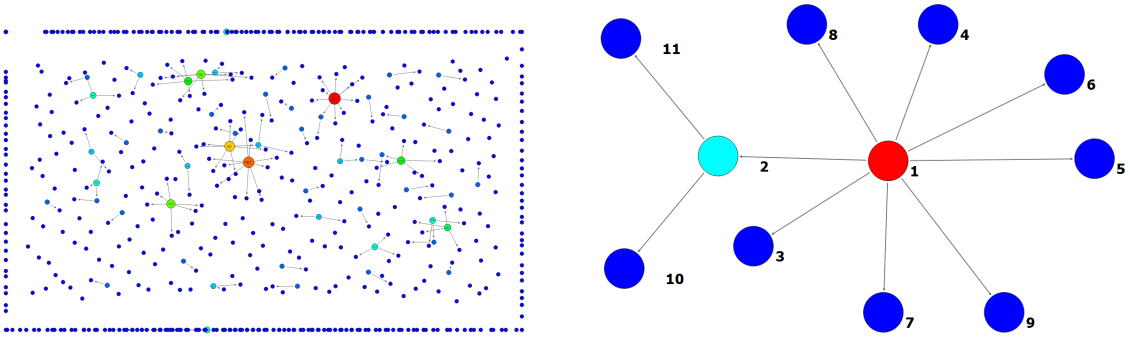
*“Impressive anticancer action on two cannabis substances”*https://t.co/YGLx59U1m5 https://t.co/19advJR7Hb

The length of the original tweet object is 3299 characters, while the length of the broadened tweet is 4708 characters.

A sample of the returned tweets, including the above, is the following:

1. “Cannabis: A miraculous banned anti-cancer drug https://t.co/Fwv4jsO8x2”
2. “Impressive anticancer action on two cannabis substances https://t.co/YGLx59U1m5 https://t.co/19advJR7Hb”
3. “The healing value of cannabis - A testimony https://t.co/vbWbnq6TNp”
4. “Wake Up: The Healing Value of Cannabis - A Testimony: The Other Side ... Read More. Wake Up ... https://t.co/uztSU8dAqC”
5. “VIDEO Greek #patients talk publicly about how they beat death with #oil #hemp. By #AndreasRoumeliotis ... https://t.co/eLUN7cjvlQ”
6. “Medical Cannabis: How It Cures Cancer, Children’s Epilepsy, and not only https://t.co/bZFXnicBeJ”
7. “It’s the most effective herbal blend for cancer. Https://t.co/k6vLpws20A EMEDI’s Herbal Cancer Blend https://t.co/Qdtb0C9JN2”

Finally, the Network Propagation module constructs the propagation representation of this fake news story on Twitter, as described in the previous section. Figure 3 depicts the diffusion network that consists of multiple propagation trees, and Figure 4 presents some interesting visualizations of network features related to centrality indices.

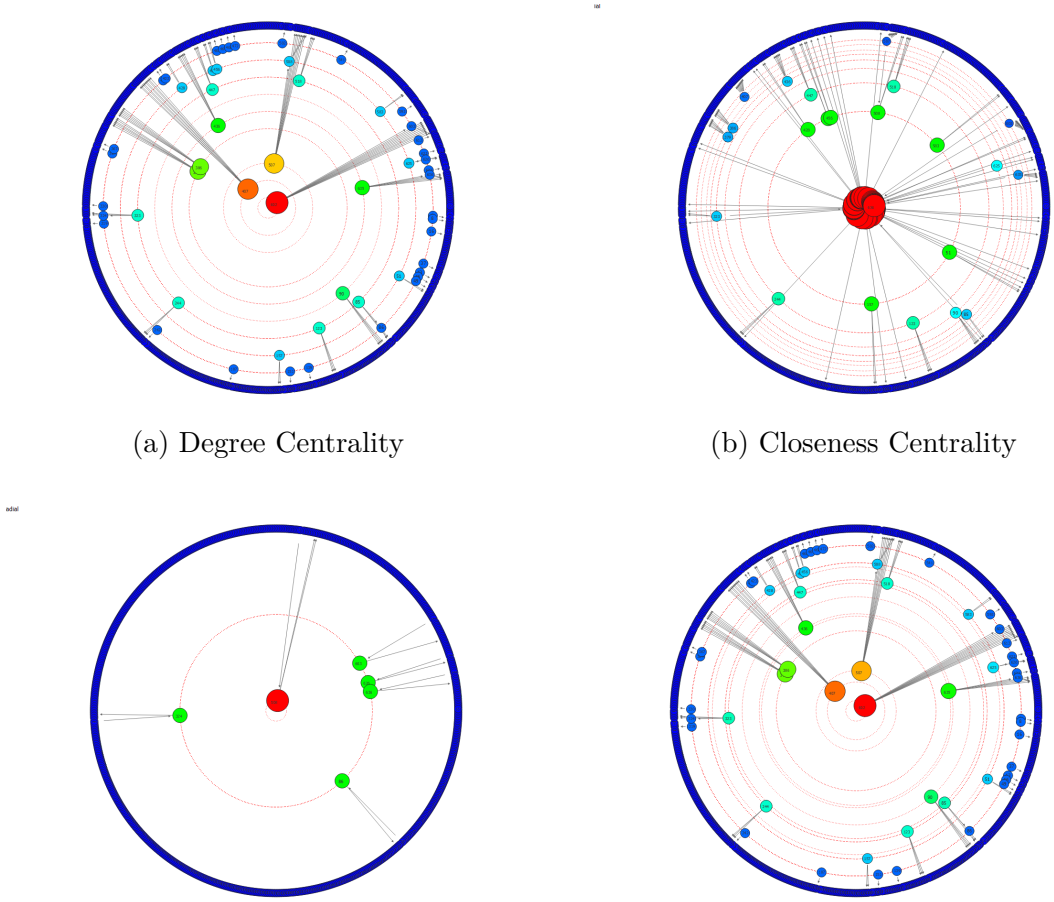


(a) Diffusion Network (b) Single Propagation Tree

Figure 4.4: Diffusion Network and Propagation Tree

At this point, the feature-agnostic dataset is complete and can be used to construct ad hoc feature datasets. For example, we consider some important and popular features used in literature approaches and show how they can be extracted from the feature-agnostic dataset using specific scripts and algorithms.

- Fraction tweets that contain user mention: This feature measures the fraction of tweets that contain one or multiple user mentions for each news story. User mentions (as well as replies) can be used to join conversations on Twitter. A mention is a Tweet that contains another person’s username anywhere in the body of the Tweet. By mentioning a user in a tweet, the user will see the tweet in their Notifications tab. To mention a user, one must type in the tweet text the ”@” symbol before the usernames she wants to mention. Our feature-agnostic dataset provides all tweets accompanied by their metadata. Specifically, we provide the field ”user_mentions” that refers to all mentions contained in each tweet. For example, the tweet with id: 788399764828282900 contains three mentions to the users with ids: 237719164, 14788231, and 25584888 (we skip the first mention since it always concerns the retweeted account) as shown in the corresponding field: ”user_mentions”:



(a) Degree Centrality

(b) Closeness Centrality

(c) Betweeness Centrality

(d) IR Centrality

Figure 4.5: Radial Layouts of Critical Centrality Indices of Diffusion Network of Fake News 215

["id": 3131326307, "id_str": "3131326307", "indices": [3, 12], "name": "kostasiz", "screen_name": "kostasiz", "id": 237719164, "id_str": "237719164", "indices": [14, 24], "name": "διογενης ο κλινικος", "screen_name": "kanekos69", "id": 14788231, "id_str": "14788231", "indices": [25, 33], "name": "Γκαθ", "screen_name": "Gath", "id": 25584888, "id_str": "25584888", "indices": [34, 45], "name": "protothema.gr", "screen_name": "protothema"]

- Fraction of tweets that contain URL: This feature measures the fraction of tweets that contain one or multiple URLs for each news story. URLs can be attached to a tweet by simply typing the URL in a tweet text. Our feature-agnostic dataset provides the field "urls" that refers to all urls contained in each tweet. For example, the tweet with id: 702242652494942209 and text: "Impressive anticancer action on two cannabis substances http://www.paraxeno.com/10745/entiposiaki-antikarkini-drasi-se-dio-ousies-tis-kannavis/" contains one url, as shown in the corresponding field: "urls": ["display_url": "paraxeno.com/10745/entiposi...", "expanded_url": "http://www.paraxeno.com/10745/ entiposiaki-antikarkini-drasi-se-dio-ousies-tis-kannavis/", "indices": [78, 101], "url": "https://t.co/iqew3Pan3W"].

- Fraction of uppercase letters: This feature measures the fraction of uppercase characters in the tweets of a news story. Our feature-agnostic dataset provides all tweet texts regarding each fake news story. Hence, one must simply count the uppercase letters in our text corpus. Specifically, we provide the

field "text" that contains the original tweet text of each tweet. For example, the tweet with id: 702242652494942209 contains only one uppercase character, as shown in the corresponding field: "text": "Impressive anti-cancer effect on two cannabis substances http://www.paraxeno.com/10745/entiposiaki-antikarkini-drasi-se-dio-ousies-tis-kannavis/".

- Propagation initial tweets: This feature measures the degree of the root in a propagation tree. Our feature-agnostic dataset provides all the propagation trees for each fake news story. For example, the degree of the root tweet (node 1) in the tree of Figure 3b is 8.

4.9 Discussion on Using PHONY

In this section, we discuss some additional PHONY features not covered in previous sections. We argue that our feature-agnostic datasets can cover the wide spectrum of features encountered in the literature.

4.9.1 Advantages and Limitations

Our system encompasses all the curated fake news stories from every source that is available through the infrastructure. In addition, it enables users to choose the type of fake news they are interested in, ensuring both the amount and variety of the generated datasets. However, the system needs to expand the available fact-checking sources and provide a mapping of fake news types that can be found among different fact-checking webpages.

To the best of our knowledge, the system provides complete representations of each fake news spreading on the Twitter network. However, this requires full access to the streaming tweets. The fact that we are interested in the propagation of Greek fake news allows us to disregard this limitation to some extent, as shown in Section 4.7.2. In the case of popular languages, a paid solution such as Twitter PowerTrack API is necessary.

Furthermore, our system manages to generate the complete fake news footprints on Twitter, overcoming the limited size and temporariness of tweets and online fake news content. First, the system broadens each tweet with the text extracted from the external links in the tweet. Alongside the tweets, our addition of the news article content provides a sizeable collection of text that ensures that no fake news footprints are lost. However, websites containing fake news often disappear after a short period. To tackle this, our system scrapes the website content of the streaming tweets continuously as they appear in the Twittersphere. This way, the social network traces of fake news can be later used by scientists when generating feature-agnostic datasets. Further, PHONY infrastructure provides up-to-date datasets since the tweet index is continuously and incrementally updated. This way, we avoid the generation of outdated datasets, considering that fake news and its spreading mechanisms evolve and adapt to avoid the current detection methods.

The complete automation of the dataset generation process is not yet possible due to limitations posed by the variety of fact-checking websites with different formats and styles and also with different fake news type labeling. For example, some labels on such websites refer to different types of fake news: Fake, False, Hoax, Fake News, Scam, Fake Quotes, Pseudoscience, Conspiracy Theories, Mostly True, Mostly False, etc. These required customizations regarding the scraping of the data are why our infrastructure currently supports a limited number of fact-checking sources. However, our system achieved full automation within these fact-checking sources, which is essential for generating uniform datasets that can be reliably assessed without relying on human annotators. Moreover,

our system does not restrict users regarding the choice of features or the way of feature measurement.

On the contrary, each feature-agnostic dataset contains fake news and its propagation footprints in the Twitter network. Therefore, using PHONY, users can automatically generate feature-agnostic datasets as a first step towards creating ad hoc feature-specific datasets. Moreover, the provided feature-agnostic datasets are suitable for further processing using specific scripts and algorithms. This way, PHONY users do not have to collect all the base-data, but only create specific scripts to extract features according to their needs. We believe that the analysis of such feature-agnostic datasets will help researchers derive new features for detecting fake news in social media.

4.10 Feature Completeness

The generated feature-agnostic datasets can adapt to all feature needs. In fact, all features encountered in the literature can be directly extracted from these datasets. The state-of-the-art fake news detection methods use features to classify social media posts. In the context of another ongoing work, we have recorded, analyzed, and implemented extraction and calculation methods for a large set of various types of existing features: text, user, web content, network structure, geolocation, multimedia, and propagation features. These features are measured based on the tweet text, tweet metadata, user metadata, retweet activity, etc.

Furthermore, the generated datasets contain temporal information, which means that each feature can be calculated as a time-series. The temporal aspect of the features is crucial because different news have vastly different footprints on Twitter, and the magnitude of these features cannot predict their truthiness by itself. The generated feature-agnostic datasets encompass all aforementioned required data, as shown in Section 4.13. However, a detailed presentation of those features falls outside the scope of this work. Many previous works have focused on these categories of features and underestimate the role of network diffusion features. PHONY infrastructure provides propagation data such that the extraction of diffusion features can be easily performed. In Tables 4.2 and 4.3, we show a representative feature subset containing network diffusion and semantic features (left column). On the middle column, the description of each feature is presented. We show which PHONY dataset component can sufficiently provide all the necessary data to calculate this feature on the right column.

4.11 Silver standard dataset noise

The datasets generated by PHONY are silver standard because they contain some degree of noise. The noise is a result of the inverted tweet index filtering for fake news-relevant tweets. This process is automated using natural language processing and keyword extraction techniques that introduce some errors in the datasets. To measure the noise in our datasets, we conducted a sampling evaluation experiment presented in the next section. Our datasets can become gold standard if we add an extra step, in which human agents will label each tweet as relevant or irrelevant to the fake news stories. Such a step would severely increase the dataset generation time resulting in outdated and partial datasets. On the contrary, this manual filtering becomes feasible in a specific application where PHONY datasets are small. In other words, after the users have firstly generated the desired datasets limited by the fake news story, time period, and source, they can filter-out manually the non-relevant tweets.

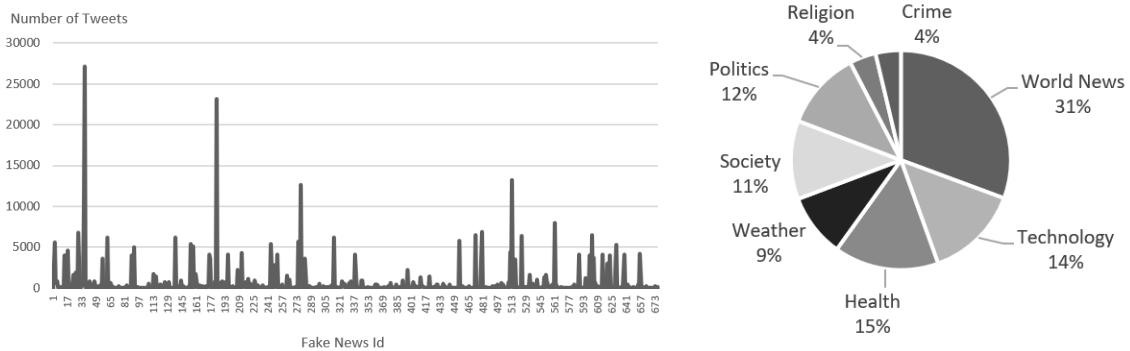
4.12 Sample Dataset Overview and Structure

To evaluate the PHONY Infrastructure, we opted to use it for generating a sample dataset. This section presents the evaluation process and the precision scores of this dataset. The generated dataset, the Greek PHONY Dataset, is an essential part of our ongoing research for developing models to detect Greek fake news content in streaming data automatically. This dataset is continuously growing; therefore, we will present only a snapshot regarding two years, an overview of the dataset characteristics, and a detailed description of the structure and the implementation process we followed.

4.12.1 Dataset Overview and Structure

We used the PHONY system to create a sample dataset including the fake news diffusion process in Greek tweets from January 2016 to April 2018. The streaming tweets were collected as described in Section 4.7.2 and stored in multiple JSON Lines files. The compressed files' size is 142.2 GB before broadening and 176.7 GB after broadening (text is broadened using the contents from URLs). Specifically, the collection contains 238 685 450 tweets posted by 1 381 799 unique users. These tweets contain 57 264 673 links to web pages. Based on the process described in Section 4.7.2, the created inverted index contains 9 237 157 terms and takes up 126.6 GB on the hard disk. These characteristics are summarized in Table 4.4. Moreover, based on the method presented in Section 4.7.3, the system created a collection of fake news regarding this period. Specifically, 677 fake news stories were crawled (by scraping *Ellinika Hoaxes* with the label "fake news"). These stories were hosted on 2 835 websites. Furthermore, the system formed and executed 677 queries against the index as described in Section 4.7.4. The queries returned 381 350 tweets, which propagated the fake news stories and were posted by 21 223 unique users. Finally, the system produced 322 265 propagation trees, while 68 fake news stories did not propagate in the Twitter network. These characteristics are summarized in Table 4.5.

The generated feature-agnostic dataset presents some interesting facts. Figure 4.6a depicts the propagation magnitude (number of tweets) of the Greek fake news stories in the Greek Twittersphere. As we can observe, the magnitude of the diffusion is highly diverse among fake news, which indicates that it is probably not an efficient detection feature.



(a) Propagation Magnitude of Greek Fake News Stories

(b) Topic-based Distribution of Top-20 Fake News Stories

Figure 4.6: Greek dataset statistics

Table 4.6 presents the 20 most propagated fake news stories in the Greek-Twittersphere, while Figure 4.6b depicts a topic-based distribution of the Top-20 Fake News Stories.

We now describe in detail the structure of the dataset and the files and formats that compose the generated dataset. The main directory of the generated dataset consists of the following sub-directories:

- fakenews_stories: this directory contains the fake news stories. A JSON Lines file named "stories.jsonl" is used for storing the ID and the URL for each fake news story and contains one fake news story object per line.
- fakenews_texts: this directory contains the texts for the fake news stories in separate text files. Each file contains the text for a specific fake news story instance. The file is automatically named according to the associated fake news story ID and a serial number, since a fake news story may have multiple instances.
- fakenews_search_results: this directory contains information on the generation of the dataset. This information can help the user understand and improve the system. A JSON Lines file named "solr_search_results.jsonl" is used for storing the fake news story ID, the Solr query that was formed by the system, and the number of tweets that satisfied the query. The file contains a search results object per line.
- fakenews_tweets: this directory contains the broadened tweets for each fake news story in separate JSON Lines files. Each file is automatically named according to the associated fake news story ID and contains a broadened tweet object per line.
- fakenews_user_network: this directory contains the user network, i.e., the follower-followee relationships, for the users that participated in the spread of the fake news stories. A JSON Lines file named "user_friends.jsonl" is used for storing the user ID and the user's friends (followees) for each user that posted at least one retweet in any fake news story. The file contains an object for one user and the user's friends per line.
- fakenews_propagation_graphs: this directory contains the propagation information for each fake news story and each propagation type in separate JSON Lines files. The files are divided into three sub-directories, each sub-directory containing the graphs that correspond to a specific propagation type. Each file is automatically named according to the associated fake news story ID and contains a propagation graph object per line. Each propagation graph object is constructed according to the tweets and the users involved in a particular tweet sequence of an original tweet and all the associated retweets. The graph nodes represent tweets, while the edges represent the friendship relationships that tweet propagation is attributed to.

As presented in previous sections, the automatic creation of silver standard datasets involves processing and querying the entire set of streaming tweets. In order to efficiently implement our system, we are using a cloud (distributed) approach consisting of 12 virtual machines (16 cores and 32GB RAM each). We use Apache Spark to process tweet data and Apache Solr to create the incrementally updating inverted index and to execute the queries on the index. Specifically, for preprocessing streaming tweets, a Spark application was implemented in Python/PySpark. Moreover, we use NLTK3 and a language-dependent stemmer to preprocess the tweets, which are then given as input to Apache Solr. Finally, the Solr index querying is performed through HTTP.

4.13 Evaluation

In this section, we complete the evaluation process by measuring the quality of the sample dataset.

As presented in Section 4.11, the PHONY infrastructure generates silver standard datasets because the dataset generation process can lead to tweets with the wrong class label. Specifically, the step of filtering the tweets in the inverted index with the fake news keywords and the automatic tweet assignment for each fake news story is an approach that will inevitably introduce some errors (poor selection of keywords, query formulation, etc.).

The precision of the sample dataset is calculated manually by counting the number of wrongly assigned tweets. However, the size of the dataset (381 350 tweets) makes a complete manual evaluation very difficult. Therefore, we opted to use a sampling evaluation method, in which 5% of the fake news stories and 10% of the average number of tweets per story were manually evaluated. Specifically, we sampled 36 random fake news stories and 56 random tweets for each story from the Greek PHONY Dataset. To select the 36 random fake news stories (from the fakenews_texts directory), we used a random number generator between 1 and 677. For each story, using the same generator, we sampled 56 (if any) tweets (from the fakenews_tweets directory). The manual evaluation was conducted by two annotators, who classified tweets into four categories:

- R (Relevant): the tweet spreads the fake news story
- RD (Relevant-debunking): the tweet debunks the fake news story
- NR (Not relevant): the tweet is not relevant with the fake news story
- N/A: inconclusive annotators decision

Figures 4.7 and 4.8 show the precision scores per fake news story. Specifically, Figure 4.7 depicts the precision scores, where true positives are considered all relevant tweets, including the tweets that debunk a fake news story. Figure 4.8 depicts the precision scores, where true positives are considered all relevant tweets, including the tweets that debunk a fake news story. The precision score was calculated as follows:

$$P_n = \frac{R + RD}{R + RD + NR + N/A} \tag{4.2}$$

As we can observe (the results are included in the dataset under the "final sample" directory), the average precision of the dataset reached 77,56%, while 19 of fake news stories (54%) achieved precision scores higher than 90%. However, some fake news stories achieved significantly lower precision scores. The main reason for this is that the web page crawling step failed to collect useful information about the fake news story. For example, the story with id "005" has two crawled versions ("005_3" and "005_06") where the system has crawled only the name of the website. As a result, the system recognized the specific website name as a keyword, and due to the website's popularity, PHONY collected many tweets that mention it, however, regarding irrelevant stories. We could enhance the crawling process by making a list of website names and excluding them from the keywords used for querying the inverted index. Moreover, we observe that stories regarding international politics and military confrontations (i.e., stories "090" and "189") achieve lower precision scores. Here, the keywords extracted from the pages are not specific enough to capture the specific story, resulting in tweets related to the topic area but not the specific news story. This problem could be addressed in the next version of the PHONY infrastructure by introducing more strict time constraints when querying the inverted index to collect tweets closer to the specific story.

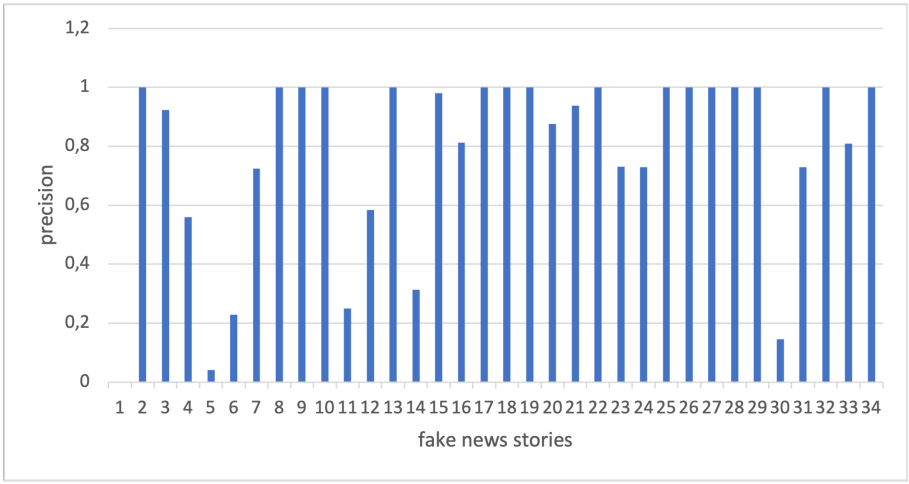


Figure 4.7: Precision for each fake news story (relevant and relevant-debunking tweets)

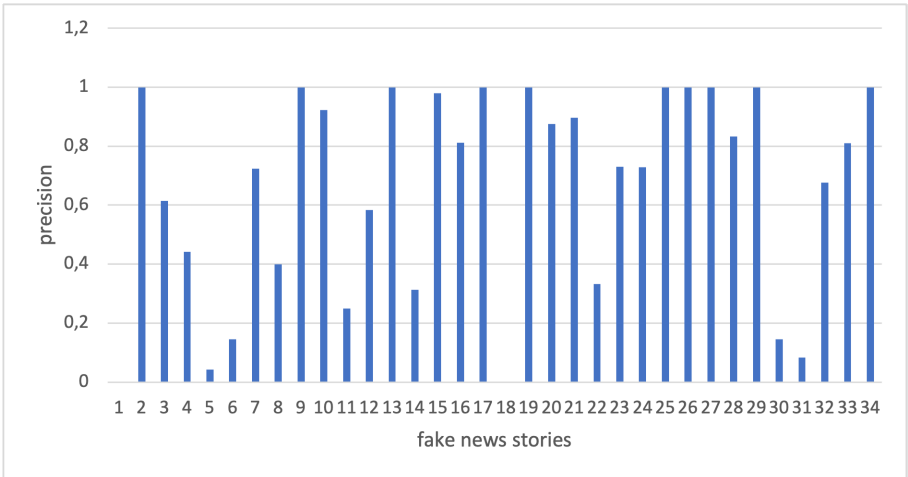


Figure 4.8: Precision for each fake news story (only relevant tweets)

4.14 Conclusion

The development of efficient algorithmic solutions for detecting fake news in online social networks requires complete, up-to-date, and flexible training datasets. In this chapter, we described PHONY, an infrastructure for automating the generation of feature-agnostic datasets. These datasets contain fake news and their propagation footprints in the Twitter network based on the fake news stories provided by curated fact-checking websites. Such datasets that are uniform and updatable can significantly contribute to developing effective fake news detection models since simple scripts and algorithms can directly extract all features encountered in the literature.

Network Features	Description	Required Data
fraction of low-to-high diffusion and fraction of high-to-low diffusion	Propagation from a user with lower influence (number of followers) to a receiver with higher influence. This kind of diffusion is a phenomenon that is seen much more frequently when the information is true.	fakenews_user_network
fraction of nodes and edges in the largest connected component of the diffusion network	Captures the longest diffusion chain of a news story in the propagation tree.	fakenews_propagation_graphs
fraction of nodes and edges in the largest connected component of friendship network	Captures the longest user engagement tweeting about a news story.	fakenews_user_network
avg. degree of nodes, avg. clustering coefficient, density, median in-degree, median out-degree, max. degree in the largest connected component of diffusion network	Structural properties of the diffusion network.	fakenews_propagation_graphs
avg. degree of nodes, avg. clustering coefficient, density, median in-degree, median out-degree, max. degree in the largest connected component of friendship network	Structural properties of the friendship network.	fakenews_user_network
fraction of nodes without incoming edges in the friendship network	Captures users that do not have any followers tweeting about the news story.	fakenews_user_network
fraction of nodes without outgoing edges in the friendship network	Captures users that do not have any followees tweeting about the news story.	fakenews_user_network
fraction of isolated nodes in the friendship network	Captures users who do not have followers or followees tweeting about the news story.	fakenews_user_network
fraction of nodes without incoming edges in the diffusion network	Captures users that started a propagation tree about a news story.	fakenews_propagation_graphs
fraction of nodes without outgoing edges in the diffusion network	Captures users who participated in the propagation of a news story, but no user continued the propagation under their influence.	fakenews_propagation_graphs
fraction of isolated nodes in the diffusion network	Captures users who started a propagation tree about a news story, but no user continued the propagation under their influence.	fakenews_propagation_graphs
initial propagation tweets	Captures the number of source-original tweets regarding the news story.	fakenews_propagation_graphs
max. propagation subtree	Captures the length of the biggest propagation chain of a news story.	fakenews_propagation_graphs
avg. propagation depth	Captures the average depth of the propagation of a news story.	fakenews_propagation_graphs
avg. depth to breadth ratio	Captures shape of new story diffusion.	fakenews_propagation_graphs
ratio of new users	Diversity/novelty of users engaged in the propagation of a news story.	fakenews_propagation_graphs
ratio of original tweets	Captures how captivating, engaging and original is the conversation about a news story.	fakenews_tweets
early burst volume	Captures the magnitude of the bursting tweets about a news story. It is the volume of tweets in the first spike of the propagation.	fakenews_tweets
periodicity of external shock	Captures the time periodicity of subsequent bursts of tweet volume regarding a news story.	fakenews_tweets
users avg. influence	num of followers/num of followees	fakenews_user_network

Table 4.2: Network Features

Semantic Features	Description	Required Data
LDA topic vectors	Captures topic vectors that signal fake/factual news.	fakenews_tweets
Named entity recognition (NER)	Captures the snippets in a text that are mentions of real-world entities	fakenews_tweets
Named entity linking (NEL)	Captures and annotates a potentially ambiguous entity (from NER) with a link to a knowledge graph node describing a unique entity (e.g. http://dbpedia.org/resource/Paris)	fakenews_tweets
Word embeddings	Captures relational similarities of the textual information of tweets.	fakenews_tweets
Absurdity (binary)	Captures the likelihood of imbalances between concepts, as well as contextual imbalances. Named entity resolution vector (ER) and named entity links (NEL) are combined by intersecting them: $NEL \cap NE = 0$ no satire, 1 satire.	fakenews_texts
Humor (binary)	Captures the semantic relatedness between the first and last article sentences that indicates the probability of an article being humorous.	fakenews_texts
Link prediction (semantic proximity)	Captures the probability of a fact to be real or fake by predicting the existence of an edge connecting the claims in a knowledge graph.	fakenews_tweets

Table 4.3: Semantic Features

Number of tweets in Tweet Index	238,685,450
Number of unique users in Tweet Index	1,381,799
Number of external links in Tweet Index	57,264,673
Number of terms in Tweet Index	9,237,157
Number of terms in Tweet Index	126.6 GB

Table 4.4: Tweet Index Characteristics

Fake News Stories	677
URLs of Fake News	2,835
Tweets propagating Fake News	381,350
Unique Users propagating Fake News	21,223
Propagation Trees of Fake News	322,265
Fake News Stories that did not propagate	68
Avg. tweets per Fake News	562.58
Max. tweets per Fake News	27,072
Min. tweets per Fake News	0
Median tweets per Fake News	38

Table 4.5: Greek Dataset Characteristics

Fake News Story	Num. of Tweets
Putin gets back the recognition of Skopje	27072
Worrying: Artificial Intelligence Experiment 'Got Away' - Robots Start Speaking in Their Own Language	23135
Shock study by BofA and The Mises Institute	13189
Cancer treatment with camphor	12661
Photos from Crete: One meter of snow in Anogia	10286
The stray from Greece are going here..	7971
Survival increases if cancer patients do not receive chemotherapy	6889
Descendant of Nazi collaborators with fake resume Foreign Minister Kojias!	6725
Unprecedented: Ramadan compulsory in French schools otherwise students will not be able to take exams!	6501
Saint Paisios: There will be a war between Russia and Turkey	6493
Schäuble confession on hidden camera: In Greece everything is predefined	6397
Why F-16 Upgrading Program in Greece is Paranoid and Dangerous for Greece	6210
Faced with life imprisonment, perhaps death penalyt the 19-year old for cocaine	6209
Stop chemotherapy	6171
Banks are closed again!	5779
Trafficking in Turkey	5663
Teacher in Lesvos decides to dress with skirts sixth grade boys	5555
China releases footage from bases on the Moon! (video)	5422
Only eight of the 27 EU member states sign the declaration in Tallinn (video)	5355
The coldest Easter of recent years is coming	5291

Table 4.6: Top-20 Most Propagated Fake News Stories

Chapter 5

KNOwDE: Knowledge-based Recommendations for Data Exploration

KNOwDE is a system that tackles the data exploration problem in the light of generating efficient knowledge-based and data-based recommendations while providing relevant data insights to the user. The user can interact with the system by selecting alternative keywords and related entities with generalization or specialization relations. Moreover, KNOwDE provides data object recommendations and two Data Graph visualizations based on the selected alternative keywords that connect the most critical and associated dataset objects. Currently, KNOwDE is implemented on top of the CORDIS dataset and uses DBPedia and ConceptNet as knowledge bases. To showcase KNOwDE, we present the system's user interface using a thorough example.

5.1 Introduction

Over the last years, data growth and availability have increased the need for efficient data exploration systems. The size and diversity of available datasets, their continuous collection and generation by systems, sensors, scientists, organizations, and the online content's growth result in rich and heterogeneous content. Hence, the volume and complexity of available data make efficient data exploration systems more and more crucial.

Data exploration encompasses a wide range of sub-topics, such as data storage and access, user interaction with data systems, relevance feedback recommendations, user interest profiling, linked data resources, query translation - modification - answering techniques, etc. Modern multi-aspect data exploration systems comprise such systems.

This chapter presents the KNOwDE system that can be used in such data exploration systems as a first-level functionality. We assume that users aim to extract knowledge from data without precisely knowing the data structure and contents. Thus, we assume that users use intuitive keywords and key-phrases in this process as they would do in a common internet search. Our system tackles the data exploration problem in the light of generating efficient knowledge-based and data-based recommendations and providing relevant data insights to the user. Hence, users pose natural language queries and use our recommendations to strive their queries in the right direction.

KNOwDE's main advantage is that it does not require for users to have prior knowledge of any details about the database structure or the query language. That is a useful feature both for information consumers as for domain experts. When building a data search interface, systems usually face the problem of natural language ambiguity. Moreover, such

systems must deal with problems like domain-dependent entities, keywords, and concepts.

In the context of this work, we use recommendations based on knowledge bases (KBs) and a Graph extracted from the database (Data Graph) to assist users unsure how to form a correct query. These recommendations help users form queries relevant to their interests and the database. KNOwDE provides alternative keywords extracted from knowledge bases, hence semantically related to the original ones to capture the actual context of the users' interests. The user can interact with the system by selecting these alternative keywords.

This way, we achieve a kind of query expansion that broadens the user's search vocabulary with relevant terms contained in the database. However, synonymy-based and directly-related recommendations with the original keywords may not be sufficient for users who are not experts in the database's terminology. For this reason, our method extends these alternatives and offers further recommendations based on generalization and specialization relations.

To rank the keyword recommendations, KNOwDE leverages the semantic relations of keywords, key-phrases by constructing and mining Keyword Graphs, which are the graph representations for each keyword and its related alternatives.

Moreover, KNOwDE uses the selected keywords and queries a data index to generate data object recommendations. To rank the data object recommendations, we use a frequency-based and a PageRank-based technique. The PageRank technique requires constructing a Data Graph, hence the graph representing the database objects and the relations derived by the database structure.

Alongside the keyword and data objects recommendations, KNOwDE provides two Data Graph views based on the selected alternative keywords, namely the Subgraph S and the Steiner tree, that connect the most critical and associated database objects. This visualization enables users to acquire a more comprehensive look at the results and helps them to interact with the recommended objects and also objects related to them.

Summing up, KNOwDE provides a semantic exploration of the database, where the concepts in the user query are matched with the concepts in the database with the help of knowledge bases that provide alternative keywords and key-phrases.

To showcase KNOwDE, we present in detail the implementation of our system, which currently integrates the CORDIS database and the DBPedia and ConceptNet knowledge bases. The KNOwDE web application is available to the research community and is continuously updated and expanded with new sources in the ongoing INODE project.

The main contributions of this work are:

- we describe in detail KNOwDE, a system for generating efficient knowledge-based and data-based recommendations for data exploration,
- we describe the KNOwDE implementation, on top of the CORDIS database and using the DBPedia [1] and the ConceptNet [2] Knowledge-bases.
- we present the KNOwDE user interface [3]

The chapter is structured as follows. Section 5.3 presents in detail the KNOwDE system. In section 5.4, we address the implementation of the system, and in section 5.5, we present the user interface and discuss further challenges and limitations of our system. Section 5.2 presents an overview of the current state of the art regarding existing methods for data exploration. Finally, we conclude the chapter in section 5.6.

[1] https://wiki.dbpedia.org/
[2] https://conceptnet.io/
[3] http://knowde.imsi.athenarc.gr/

5.2 Related Work

Query expansion, query suggestion, and query refinement are some of the most well-known query modification techniques. Query expansion is a method of expanding the user's original query to improve the information retrieval efficiency. Query refinement provides a new query based on user query history, and query suggestion recommends several queries related to the user's interests. Over the last years, knowledge-based query recommendation methods have gathered the attention of many research works. In [158], the authors use Freebase (knowledge base) for query expansion. Specifically, they use TF-IDF and Pseudo-Relevance Feedback to analyze the objects' descriptions (KG nodes). Although the method did not improve the already well-formulated queries, it significantly improved the retrieval efficiency of weak queries. In [16], the authors used DBpedia and ConceptNet and a co-occurrence metric for selecting the expansion keywords.

Moreover, some works [158, 18, 160, 11, 13, 40] use knowledge bases (Wikipedia, Freebase, DBpedia). However, they rely on relevance feedback from the top retrieved documents (both with and without manual user interaction). Relevance feedback uses the results returned from a given query and uses them (along with the user's information about whether those results are relevant) to recommend a new query. The assessed documents' content is used to adjust the weights of terms in the original query and/or to add words to the query. Relevance feedback essentially is a demanding approach regarding resources because the retrieval and analysis of top retrieved articles is required for each query.

Furthermore, these methods face severe limitations since they reinforce the system's original decision by making the expanded query more similar to the retrieved relevant documents. Furthermore, knowledge-based approaches [158, 16, 18, 160, 11, 13, 12, 40] face another critical issue: they do not provide personalized recommendations based on user history queries. Given that user queries are usually small (number of terms) and that the natural language is characterized by ambiguity (the same word with different meanings and various words with the same or similar meanings), user history queries can shed light on the user's actual topics of interest, resulting in a more explicit recommendation context. Finally, in [169], the authors propose a method that uses personalized recommendations; however, this method is based on relevance-feedback and folksonomy data.

In this work, we used query suggestion and expansion in the sense that the original query is modified or extended with alternative and related keywords and key-phrases. Moreover, browsing these keywords and interacting with graph visualizations assist users in exploring the data semantically. Furthermore, indexing only a small part of the dataset reduces the cost of relevance feedback. The use of knowledge bases captures the context of small queries, a task in which language-based approaches would fail. On the other hand, only relying on external data such as knowledge bases would result in recommendations that may not exist in the dataset, hence disorient the user rather than help him focus and explore the specific dataset.

5.3 The KNOwDE System

KNOwDE is an essential part of our ongoing research for developing systems for generating efficient knowledge-based and data-based recommendations for data exploration, assisting users to form efficient queries.

5.3.1 Overview

KNOwDE is addressed to users without prior knowledge of the type, size, and data structure in the database. Therefore, we assume that KNOwDE users tend to use queries that seem like everyday internet search keyword lists (e.g., "science museum") and key-phrases (e.g., "what dinosaur has 500 teeth").

Our system provides two types of recommendations: (a) alternative keywords and key-phrases, and (b) related data objects. Alternative keywords can help expand users' vocabulary, reduce the effect of natural language ambiguity, and help users focus on domain-dependent entities and align their query with the objects in the database. The related data objects can help users overcome the boundaries of term-based queries and help them navigate through the data to discover interesting objects and relations.

Both the keyword and the data object recommendations assist users to interactively explore the database based on semantic assistance and graph visualization.

Therefore, our system must:

- expand the keyword query, by recommending relevant alternative words and phrases that exist in the database, and
- find the corresponding data objects and expand them with other related to them

KNOwDE system encompasses five main units, namely the:

- **Keyword Extraction and Preprocessing Unit**, which is responsible for extracting and preprocessing the keywords and key-phrases from the users' queries. To make the best possible comprehension of the user's query, this unit uses syntax analysis to break the query into keywords and key-phrases.
- **Data Preprocessing Unit** that utilizes the database to construct the Data Inverted Index and the Data Graph to assist both the alternative keyword and the related object recommendation.

  The **Data Inverted Index** is a quick way to cross-check users' queries and keyword selections to the database. It utilizes all available textual data in the database to map all keywords to the data objects that they appear in. It filters the candidate recommendations extracted from the knowledge bases to recommend keywords and key-phrases that the database comprises. It also extracts all data objects related to the recommended keywords and provides the data object recommendations.

  The **Data Graph** is a graph representation of the database objects and their relations and aims to provide a common basis for ranking the data object recommendations. It also provides the data relations for expanding the recommended objects with related ones.

  Due to database diversity, this unit is not fully automated and adapts to the specific database. Some manual steps are required to derive the table joins and ontology schema parts for building the Data Graph with helpful information to the user.
- **Knowledge Base Unit** that generates the alternative keyword recommendations by mining knowledge bases. Its main goal is to find semantically related items and rank them based on their relevance. First, customized HTTP requests to knowledge base APIs extract the semantically related items as well as their metadata and form a set of semantically related items. These items are filtered through the inverted index to filter out the ones that do not exist in the database. To rank the remaining keywords, we use ranking algorithms applied on the **Keyword Graphs**, which are graph representations for each keyword and its related alternative items

and metadata. To help users navigate through the data more effectively, we extend the recommended keywords with more general and more specific entities using knowledge bases.

- **Inverted Index Unit**, which is a multi-purpose component in the recommendation process. Its main goal is to provide a quick reference to existing keywords in the database to avoid recommending alternative keywords that are absent from the database. It also locates the data objects that contain the keywords.

- **Data Graph Unit** that uses the Data Graph to rank the data objects that contain the keywords selected by the user. To enhance our recommendations this unit constructs two subgraphs of the Data Graph which can help users deepen their understanding of the data objects by extending the top-ranked data objects:

  - the Shortest Subgraph S represents the "common neighborhood" that the top-10 data objects share in the Data Graph. This subgraph extends the top-ranked data object recommendations by adding the objects that belong to the shortest paths (S) that link every two top-ranked objects.
  - the Steiner tree of S provides a more concise data object recommendation by extracting the least-cost connected subgraph of S. This subgraph comprises some objects that may not be directly related to the user's initial query. However, we assume that if an object belongs to the Steiner tree, then the likelihood of it being of interest increases.

  These subgraphs provide a data exploration basis for the dataset and support the user in formulating the right queries.

To sum up, the process is as follows: a user forms an initial keyword query, and the system recommends alternative keywords extracted from knowledge bases that also exist in the database. Then, the user can select the keywords that fit her needs. Based on the user keyword selection, the system recommends the top-10 data objects, the Shortest Subgraph S of the Data Graph, and the Steiner tree of S. The user may continue this process iteratively until she finds queries that seem to describe her goals best.

5.3.2 System Architecture

Figure 5.1 depicts the KNOwDE architecture. This section describes the processes taking place in the five main units mentioned in the previous section. Figure 1 depicts the overall architecture of the implemented KNOwDE system. **Data Preprocessing Unit** The Data Preprocessing Unit is a unit that analyses and processes the dataset that is going to explore. Due to dataset diversity, this unit is not fully automated. It adapts to the specific database to provide the two background infrastructures necessary for implementing the method: (a) a Data Inverted Index of the textual data and (b) a Data Graph.

Inverted Index Builder: The database to be explored may include tables and fields that go beyond the requirements for the recommendation. Hence, it is essential that we decide what database objects are to be recommended. For example, the main data objects-to-recommend for a movie dataset could be the "movie titles".

Datasets can consist of structured and unstructured data, numeric, textual, in one or more files or databases, etc. In any case, to build the Data Inverted Index, the Inverted Index Builder needs to analyze textual data. Therefore, we have to manually select the tables and the fields that contain textual data, e.g., "movie review", to index the data objects. Hence, the constructed Data Inverted Index maps keywords to the data objects in which they appear.

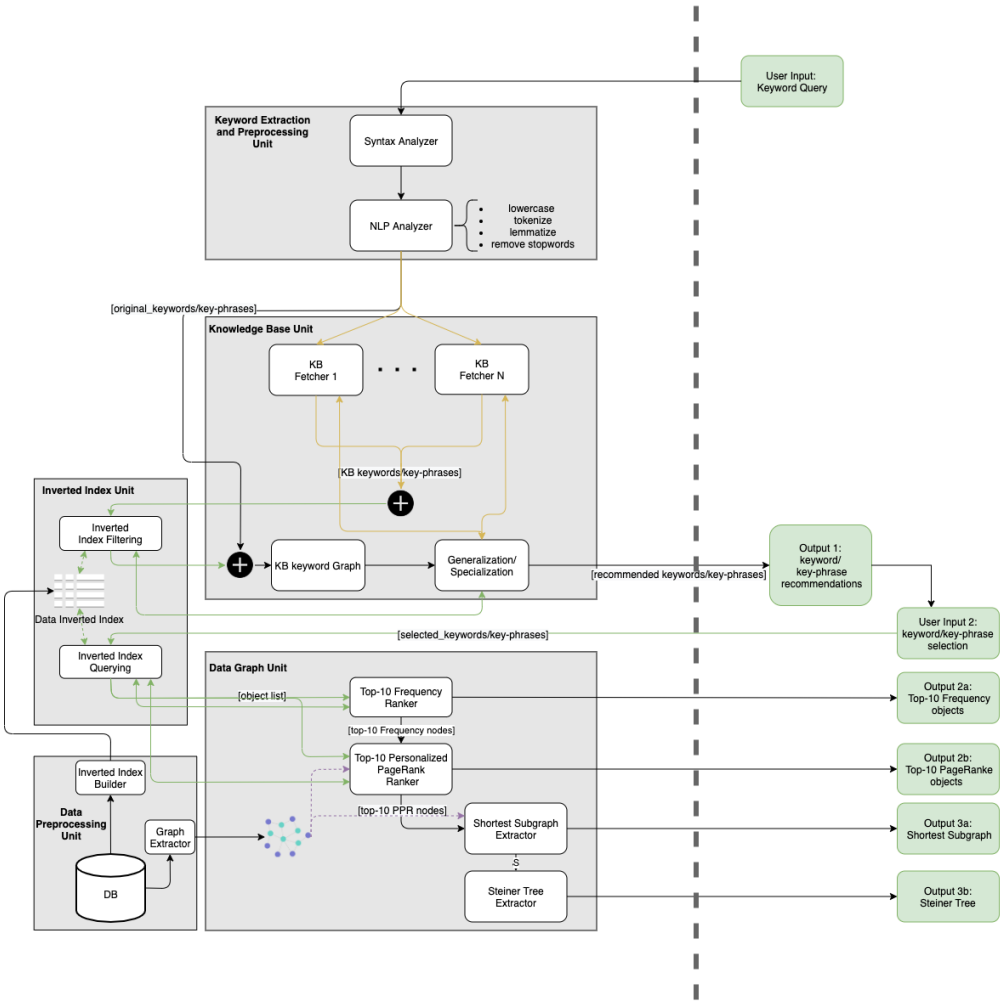


Figure 5.1: KNOwDE System Architecture

Graph Extractor: Database items are linked through various relations. Our system uses these relations to recommend additional objects by providing two graph recommendations: the subgraph S and the Steiner tree of S. For example, these subgraphs, apart from the movie titles, would also contain directors' names and producers that are related to the recommended movies. Hence, once we have chosen which items are the objects for the recommendation and which relations are important, the Graph Extractor creates that Data Graph that comprises these objects and relations.

Keyword Extraction and Preprocessing Unit This unit receives a query input from the user, namely a natural language keyword text.

The **Syntax Analyzer** breaks the query into keywords and key-phrases. For more accurate results, the system allows users to define a key-phrase using double quotes explicitly. The Syntax Analyzer outputs a list of key-phrases and keywords that the NLP Analyzer then preprocesses.

Specifically, for each keyword or key-phrase, the **NLP Analyzer** performs the following steps: lowercase text, tokenization, stopword removal, lemmatization. The Keyword Extraction and Preprocessing Unit outputs a list of preprocessed keywords and key-phrases. **Knowledge Base Unit** This unit generates the alternative keywords by

mining the knowledge bases. The recommended keywords are obtained through a two-step process of extracting and classifying the nodes of a graph (KB keyword graph) that we create based on the two knowledge bases' items. Specifically, for each keyword or key-phrase from the Keyword Extraction and Preprocessing Unit, the **KB Fetchers** extract the related knowledge base concepts (single and multi-term) using the corresponding knowledge base APIs. Moreover, the collected concepts, items, and metadata are filtered through the Data Inverted Index to keep only entities existing in the dataset. Then, they are integrated with the original keywords and key-phrases from the user query.

The process continues in the **KB Keyword Graph** unit, which constructs a graph based on the entities found in our dataset. Specifically, for each item (keyword or key-phrase) from the original user query, it creates a graph where the root node is this item, and an edge connects the root with each knowledge base entity. Moreover, extra edges connect the knowledge base entities with their metadata. The original query items and the extracted entities included in the Data Inverted Index are represented as the Keyword Graph nodes, and in the case of multiple instances of an entity, the KB Keyword Graph merges them into one node.

Finally, this unit computes the personalized PageRank scores for each node using as seed nodes the common nodes between the entities of the knowledge bases, and outputs the nodes with the top-5 scores.

The unit's final component is the one that extends the suggested keywords to semantically more general and more specific entities. This functionality helps the user to navigate more efficiently through our data space, enabling him to search for data based on relevant keywords and errors. Specifically, the **Generalization/Specialization** component queries the API of the knowledge bases for each recommended entity and gets the top 5 items related to the entity by a parent-relation (generalization), e.g. "IsA" or by child-relation (specialization), e.g. "Has". These extra entities are then filtered through the Data Inverted Index, and the results are integrated with the recommended entities to form the first output of the system.

Inverted Index Unit The Inverted Index Unit is used multiple times in the recommendation process. However, the system queries the index mainly in two phases:

1. To filter the keywords extracted from the knowledge base. The **Inverted Index Filtering** component checks every keyword against the index to eliminate the keywords that do not exist in our dataset.

2. To find the data objects containing the user's keywords. Specifically, after the user selects a set of the recommended keywords and key-phrases, the **Inverted Index Querying** component forms a query using the "OR" operator to combine these selections. Finally, it runs the query using a scoring metric of the frequency of the indexed keywords. Hence, it finds the objects that contain at least one selected keyword or key-phrase and the frequency of their occurrences.

Data Graph Unit This unit covers the Data Graph processing, based on the user's choices, to provide graph exploration recommendations. The Data Graph Unit's inputs are the data objects, which contain the selected user keywords and key-phrases, and that resulted from querying the Data Inverted Index.

The **Top-10 Frequency Ranker** component ranks the input objects based on the keyword occurrences and outputs the top-10 objects and the corresponding text excerpts found in the dataset. To find the excerpts, it performs a specific query to the Data Inverted Index.

Moreover, the **Top-10 PageRank Ranker** component ranks the input objects based on the personalized PageRank algorithm. As seed nodes, we use the top-10 objects based

on frequency. This component also outputs the corresponding text excerpts found in the dataset by querying the Data Inverted Index.

Furthermore, the **Shortest Subgraph Extractor** extends top-10 nodes ranked by the Frequency and the PageRank Rankers with the nodes on all shortest paths connecting them. The component constructs the Data Graph subgraph formed by all shortest paths (S) that link every two nodes in the top-10 nodes using the networkx [4] shortest path algorithm for every pair of nodes. This process results in a bigger nodelist that contains all intermediate nodes in the shortest paths. If the selected keywords result in a single node, this unit generates the node's Ego-Network. Our intuition behind using the subgraph S is that it can give the "common neighborhood" that the top-10 nodes share in the Data Graph. It is not the most concise nor the most verbose representation of this neighborhood.

Finally, the **Steiner Tree Extractor** returns the nodes of the least-cost connected subgraph of the S graph. Specifically, the component extracts the Steiner Tree of S that links the top-10 nodes ranked by the Frequency and the PageRank Rankers. The intuition behind the Steiner Tree's use is that if a node is connecting two or more top-10 nodes in the DB graph, then this node is likely itself essential. Although some nodes may not be directly related to the user's initial query, we assume that if a node belongs to the Steiner Tree, then the likelihood of it being a node of interest increases.

5.4 System Implementation

This section describes the KNOwDE implementation, on top of the CORDIS[5] database, using the DBPedia and the ConceptNet knowledge bases. We also describe the data integration methods for the CORDIS data. Currently, KNOwDE encompasses these knowledge and data sources, however the system is designed to facilitate the addition of new sources. **The CORDIS dataset** The CORDIS (Community Research and Development Information Service) dataset and ontology comprises the data regarding the European Commission's primary source of results from the projects funded by the EU's framework programmes for research and innovation and is available through the SQL dump of the CORDIS dataset. The dump comes from a PostgreSQL 9.5 database. Picture 3a depicts the CORDIS DB Schema. We used the CORDIS dataset so that the research projects are at the center of the exploration. Additionally, we opted to use more information regarding the research projects: the members that participate, the subject areas, and the programs they belong to. The data above are encompassed in the tables: projects, project_members, subject_areas and ec_framework_programs. Figure 5.2 depicts the architecture of the implemented KNOwDE system.

Data Preprocessing Unit In this section, we describe the Data Preprocessing Unit, which processes the CORDIS Database contents.

Inverted Index Builder: For the generation of the Inverted Index, we have had to select the table items that contain Natural Language textual data. Hence, the unit uses the "objective" field from the projects table since it encompasses the abstract of each project. The constructed Data Inverted Index maps keywords to the projects in which they appear. For example, assume an inverted index, as shown in Table I, for the field "Objectives".

We used the Whoosh [6] library for indexing 46254 projects. The inverted index occupies 404 MB on the hard disk.

Graph Extractor: We opted to use the relationships that link the projects with

[4]https://networkx.org/

[5]https://ontology-documentation-inode-cordis.s3-eu-west-1.amazonaws.com/index.html

[6]https://whoosh.readthedocs.io/en/latest/quickstart.html#a-quick-introduction

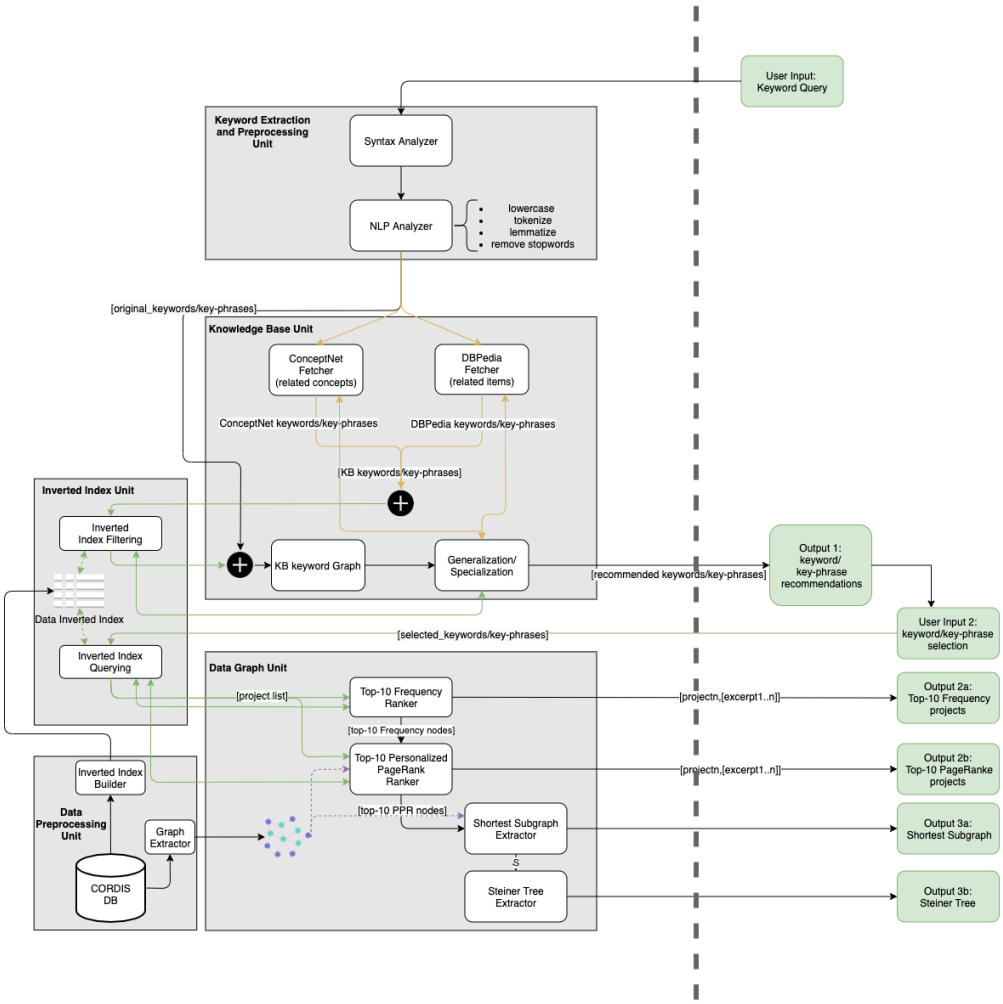


Figure 5.2: KNOwDE Implementation Architecture

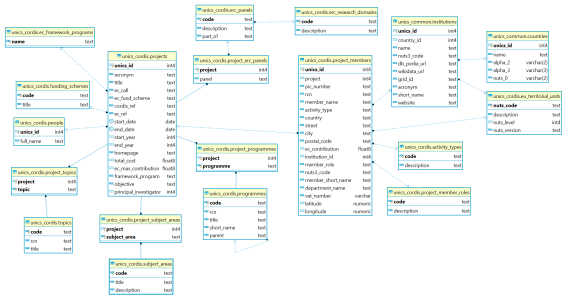
Keyword	**Objectives**
solar	[(project=1, freq=2), (project=2, freq=5), (project=3, freq=1)]
car	[(project=2, freq=1), (project=3, freq=2)]

Table 5.1: Inverted Index Example

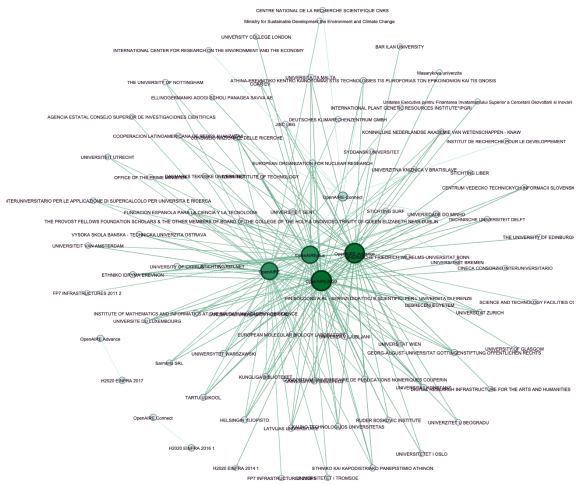
other data objects that could offer users an understandable but not overly detailed insight of the research projects. These data objects are the project members, the subject areas and the programs they belong to.

The **CORDIS Data Graph** is the induced graph based on the dataset. Thus, we selected the tables containing the nodes to include in the graph (projects, members, subject areas, and framework programs) from the CORDIS DB. Specifically, we used the records from the database tables mentioned above as nodes, and the table joins between them

as edges (project-framework_program, project-member, project-subject_area). Moreover, we have processed these table joins to produce the CORDIS edge list as a simple 19,3 MB CSV file containing 331075 edges and 130335 nodes. Picture 3b depicts a part of the CORDIS Data Graph related to the OpenAire projects, where the size and color of nodes are according to their PageRank scores. Unfortunately, complete visualization of the CORDIS Data Graph is impossible due to its' large size. To help readers compare the graph visualizations, we provide Figure 4 that depicts the graph outputs of KNOwDE for the keyword "OpenAIRE". **Keyword Extraction and Preprocessing Unit** This

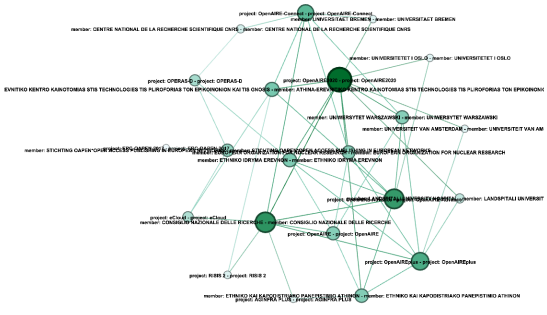


(a) SIRIS CORDIS DB schema

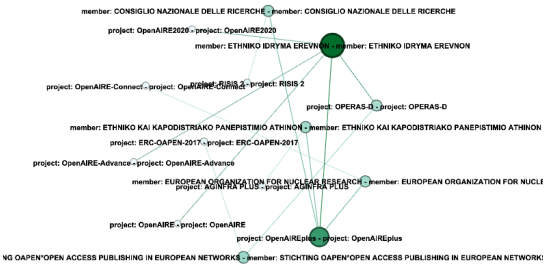


(b) OpenAire CORDIS subgraph

Figure 5.3: CORDIS Database and Data Graph



(a) Shortest Path Subgraph for keyword "OpenAIRE"



(b) Steiner Tree for keyword "OpenAIRE"

Figure 5.4: Output Visualizations for keyword "OpenAIRE"

unit receives a query input from the user and outputs a list of preprocessed keywords

and key-phrases as described in section 5.3.2. Currently, the system supports keyword and key-phrase queries. Specifically, the system allows users to define a key-phrase using double quotes explicitly. Moreover, users can form multi-keyword/key-phrase queries by adding spaces between the keywords/key-phrases. For example, for the query *"computer science" and surveillance*, the unit outputs a key-phrase ("computer science") and a keyword ("surveillance").

Knowledge Base Unit This unit generates the alternative keywords by mining into two knowledge bases: the ConceptNet and the DBPedia. Specifically, for each keyword or key-phrase from the Keyword Extraction and Preprocessing Unit:

- The **ConceptNet Fetcher** extracts the related ConceptNet concepts (single and multi-term) using the ConceptNet API, and
- The **DBPedia Fetcher** extracts the related DBPedia entities (single and multi-term) using the DBPedia Lookup API. The DBPedia Fetcher also stores the Class Label and Category (DBPedia metadata) for each entity.

Moreover, the collected concepts, items, and metadata are filtered through the Data Inverted Index to keep only entities existing in the dataset. Then, they are integrated with the original keywords and key-phrases from the user query.

The process continues in the **KB Keyword Graph** unit. For each item, keyword or key-phrase, from the original user query, the unit creates a graph where the root node is this item, and an edge connects the root with each ConceptNet and DBPedia entity. Moreover, extra edges connect the DBPedia entities with their Class Labels and Categories. The original query items and the extracted entities included in the Data Inverted Index are represented as the Keyword Graph nodes, and in the case of multiple instances of an entity, the KB Keyword Graph merges them into one node. Figure 2 depicts an example of the keyword graph constructed based on a two-keyword query.

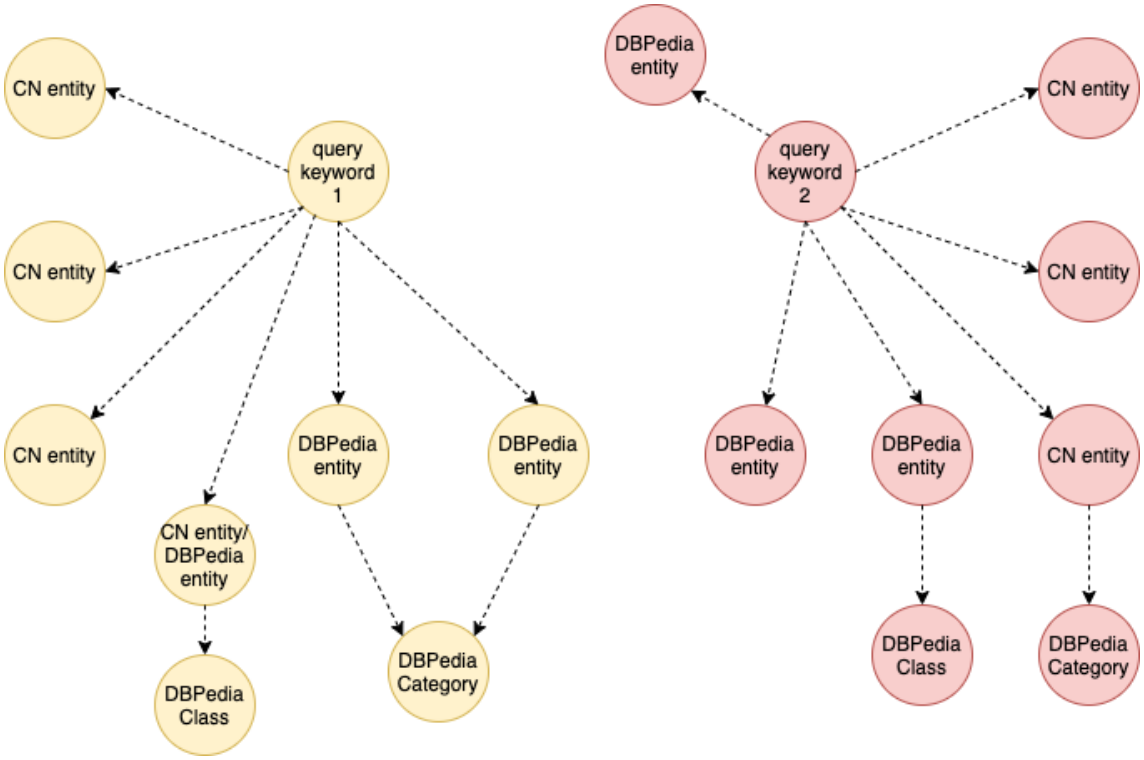


Figure 5.5: Keyword Graph

Finally, this unit computes the personalized PageRank scores for each node using as seed nodes the common nodes between the ConceptNet and the DBpedia entities, and outputs the nodes with the top-5 scores.

The **Generalization/Specialization** component calls the ConceptNet API for each recommended entity and gets the top-5 items related to the entity by the relation "IsA" either as parent (generalization) or as child (specialization). These extra entities are then filtered through the Data Inverted Index, and the results are integrated with the recommended entities to form the first output of the system.

Inverted Index Unit Our system queries the Inverted Index by using the Whoosh library for Python, as described in Section 5.3.2.

Data Graph Unit The Data Graph Unit's inputs are the projects containing the selected user keywords and key-phrases resulting from querying the Data Inverted Index.

The **Top-10 Frequency Ranker** component ranks the input projects based on the keyword occurrences and outputs the top-10 projects and the corresponding text excerpts found in the CORDIS DB, as presented in section 5.3.2.

Moreover, the **Top-10 PageRank Ranker** component ranks the input projects based on the personalized PageRank algorithm. As seed nodes, we use the top-10 projects based on frequency, as presented in section 5.3.2.

The **Shortest Subgraph Extractor** and the **Steiner Tree Extractor** extend the top-10 projects ranked by the Frequency and the PageRank Rankers with the nodes on all shortest paths connecting them (Subgraph S) and the steiner tree, as presented in section 5.3.2. S contains members, frameworks, subject areas, and more projects. If the selected keywords result in a single project, these units extend the project by its Ego-Network: its members, frameworks, and subject areas.

To sum up, the user - KNOwDE interaction is depicted in Table 2.

5.5 User Interface and Discussion

We developed the system as a web application available to the research community. In this section, we are presenting the KNOwDE web app[7] through a use case example and discuss the main characteristics of our method.

5.5.1 KNOwDE User Interface

Suppose a user wants to explore CORDIS data related to computer science for surveillance purposes. Hence, based on the instructions, the user types the following query: '"computer science" and surveillance' in the query box and hits 'Go', as shown in Figures 5a and 5b. The instructions follow the syntax rules presented in section 5.3.2.

Then, the app returns the recommended keywords and key-phrases, their knowledge base origin (ConceptNet and DBPedia), the related generalization and specialization recommendations, and their PageRank score. The user can now extend his query by selecting some engaging keywords (computer science, software framework, surveillance, security) and some more general/specific entities (artificial intelligence). The information mentioned above is shown in Figures 5c and 5d and is generated according to the method described in section 5.3.2.

Based on user selections, the KNOwDE extracts the top-10 related projects using the personalized PageRank (Figure 6a) and the keyword Frequency (Figure 6b). The system presents these projects, the keywords they contain, and the text excerpts they occurred in, as described in section 5.3.2. Moreover, the app returns two visualizations: the Subgraph

[7] http://knowde.imsi.athenarc.gr/

User Inputs
1. Natural language queries. The system identifies keywords and key-phrases using a specific syntax.
2. Selected keywords and key-phrases from a list of entities extracted from ConceptNet and DBPedia.
System Outputs
1. List of related entities extracted from ConceptNet and DBPedia.
2. Top-10 projects from Data Inverted Index based on selected keyword frequency.
3. Top-10 projects based on personalized PageRank from the Data Graph.
4. Subgraph S: visualization of the "common neighborhood" of the top-10 projects based on (a) personalized PageRank and (b) Frequency count.
5. Steiner tree: visualization of the "shortest connection" tree of top-10 projects based on (a) personalized PageRank and (b) Frequency count.

Table 5.2: Summary of User - KNOwDE interaction

S and the Steiner tree (Figure 6) of top-10 projects based on personalized PageRank and keyword Frequency. As described in section 5.3.2, they depict the "common neighborhood" and the "shortest connection" tree between the top-10 projects.

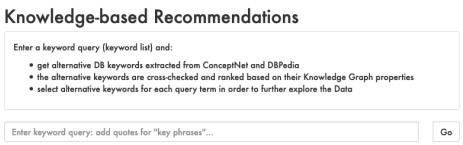
5.5.2 Discussion

The use of knowledge bases ensures the extraction of a semantically related recommendation pool, while the graph algorithms used to filter this recommendation pool of entities further enhance the recommendation efficiency. Moreover, knowledge bases are a solid background when building a recommendation system without the users' query log history and help address the cold start problem to some degree.

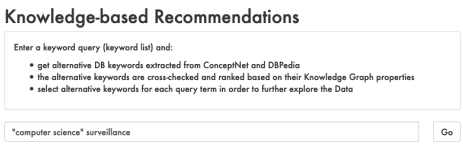
As we have tried to show in the above sections, this method can be generalized to explore other datasets. This generalization, of course, assumes that these datasets include at least some natural language data and that some relations exist between them. In the implementation that we present in this work, we adapted the method for the CORDIS dataset in the context of our participation in INODE (Intelligent Open Data Exploration) research project [8]. In this project, KNOwDE could be used as an initial component within a services pipeline to help users explore and interact with open data.

In the CORDIS implementation, we used the knowledge bases through their available APIs. However, a local deployment on our server would significantly improve the system's speed, even though it would take up significant storage resources. A more versatile exploration of the dataset could be achieved by using other operators besides the OR to

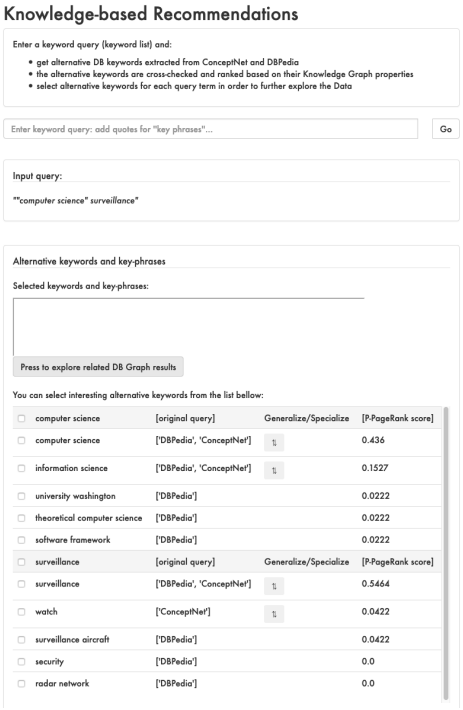
[8] https://cordis.europa.eu/project/id/863410



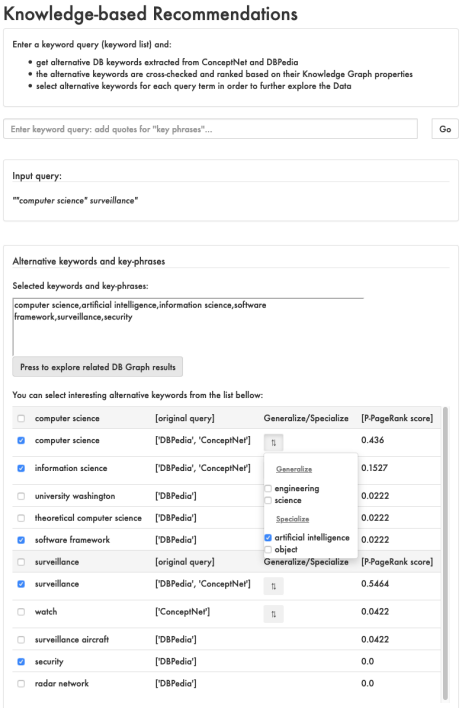
(a) KNOwDE index page



(b) User query



(c) KNOwDE recommendations



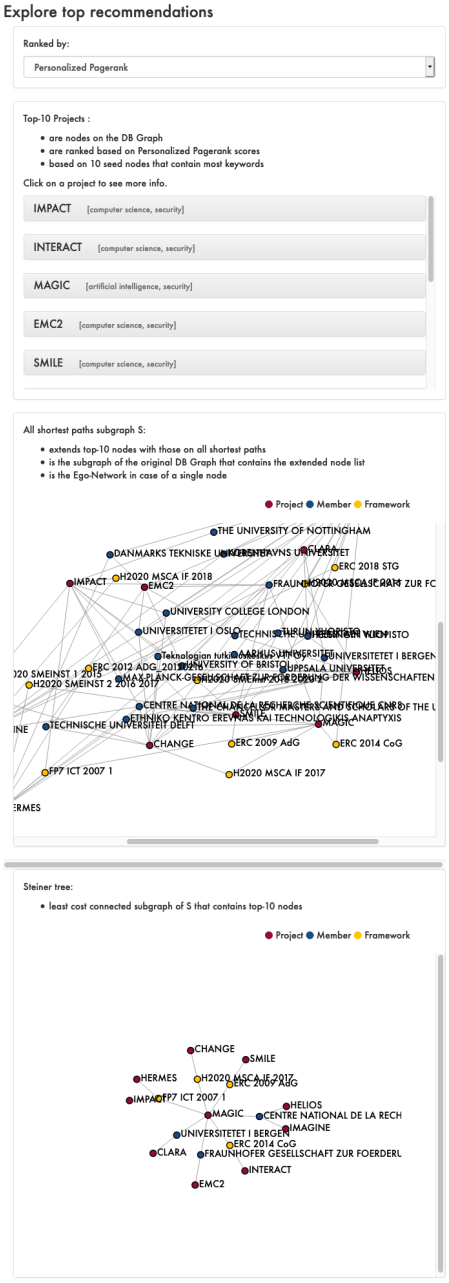
(d) User selections

Figure 5.6: KNOwDE Web Application - part1

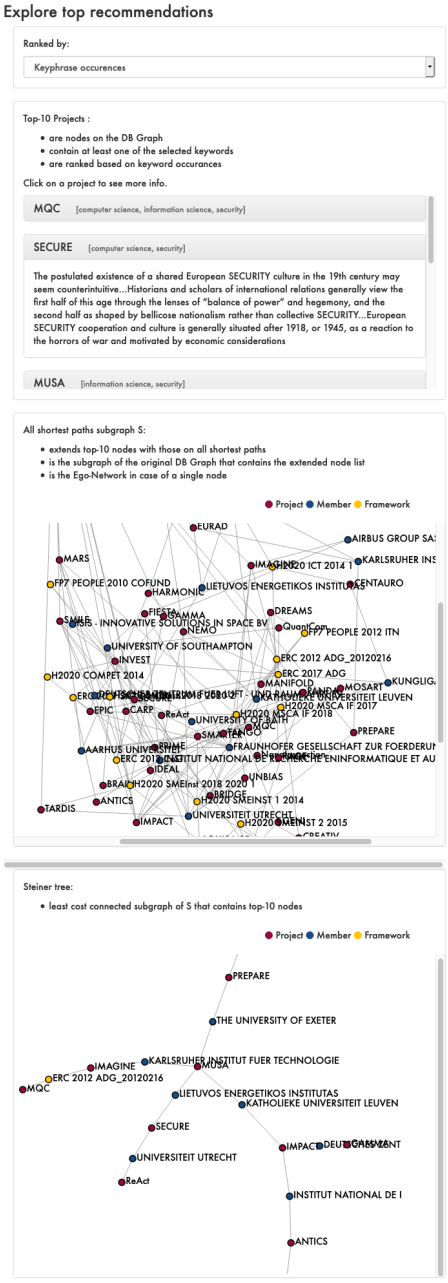
combine user-selected keywords. For example, we plan to introduce options that will allow users to exclude a group of keywords, include it all together, and combine keyword groups with different operators (AND, OR, NOT). Furthermore, we are working on enhancing user-graph interaction by enabling users to see additional information about each node of the graph (the Steiner and the S Subgraph) and expand it.

5.6 Conclusion

KNOwDE is a method that tackles the data exploration problem in the light of generating efficient knowledge-based and data-based recommendations and providing relevant data insights to the user. It is based on keyword and key-phrase recommendations extracted from knowledge bases and data insights and visualizations based on a Graph extracted from the dataset. In this work, we also presented the implementation of KNOwDE method for the CORDIS dataset.



(a) Top-10 PageRank projects and Graph visualizations



(b) Top-10 Frequency projects and Graph visualizations

Figure 5.7: KNOwDE Web Application - part2

Chapter 6

Applications leveraging social network data for exploring datasets

6.1 Tweet-based attention for articles relevant to COVID-19

The outbreak of the coronavirus pandemic has turned us into new priorities regarding social network analytics. Specifically, we collaborated with the team developing the BIP!Finder[1] to help create a dataset that will consist of literature related to COVID-19 and will comprise a social media altmetric indicator.

6.1.1 Bip4Covid Dataset

Over the last year, many scientific articles were published, which along with the older relevant literature, can be a valuable source of knowledge to support current research processes and decisions regarding the pandemic. Such articles are published rapidly, making it very difficult to effectively explore and extract useful information from them. In this context, the team's main objective was to produce BIP4COVID19[144], an openly available dataset containing various impact measures calculated for COVID-19-related literature. These impact measures facilitate the exploration of the coronavirus-related literature, providing various indicators of scientific impact for the articles. Specifically, BIP4COVID19 comprises four citation-based impact measures (Citation Count, PageRank [97], RAM ([53]), and AttRank [68]) were calculated, as well as an altmetric indicator (Tweet Count). The selected measures cover different impact aspects of the articles. The dataset also comprises the values of the following impact measures: Influence: Citation-based measure reflecting the total impact of an article. This is based on the PageRank network analysis method. In the context of citation networks, it estimates the importance of each article based on its centrality in the whole network. This measure was calculated using the PaperRanking[2] library. Influence_alt: Citation-based measure reflecting the total impact of an article. This is the Citation Count of each article, calculated based on the citation network between the articles contained in the BIP4COVID19 dataset. Popularity: Citation-based measure reflecting the current impact of an article. This is based on the AttRank citation network analysis method. AttRank alleviates the bias against recently

[1] https://bip.imsi.athenarc.gr/

[2] https://github.com/diwis/PaperRanking

published articles by incorporating an attention-based mechanism. Popularity alternative: An alternative citation-based measure reflecting the current impact of an article. This is based on the RAM citation network analysis method that alleviates the bias against recently published articles using an approach known as "time-awareness". Social Media Attention: The number of tweets related to this article. Relevant data were collected from the COVID-19-TweetIDs[3] dataset.

BIP4COVID19 data are updated regularly and are openly available on Zenodo[4]. Since its initial launch in March 2020, 43 versions of BIP4COVID19 have been released, with Zenodo recording more than 90,000 views and 10,000 downloads.

6.1.2 Measuring attention on social media for articles related to COVID-19

Our contribution in creating the BIP4COVID19 dataset regards measuring the Twitter attention received by each article. This altmetric augments the citation-based metrics and involves measuring the number of recent tweets mentioning the scientific articles. We consider this a measure of social media attention for each literature article relevant to COVID-19.

In order to produce social media attention metrics for each literature article, we would have to follow one of the following two strategies:

1. monitor the worldwide stream of tweets or

2. perform several past search queries on Twitter historical data,

for content relevant to each scientific article.

However, these approaches are very "expensive" in terms of time resources, resulting in valuable attention metrics staying unavailable for a significant amount of time. This is because, on the one hand, the cost of accessing the entire stream of tweets is high, and on the other hand, the number of articles published is so large that multiple searches on historical tweets would yield some but not all relevant tweets due to restrictions posed by the Twitter API.

To avoid these outcomes, we followed an alternative to the first approach. Specifically, we used an existing dataset with tweets related to COVID-19. However, these tweets do not necessarily refer to any scientific article. This reduces the volume of tweets for mining references to covid19-related articles. Figure 1 depicts the overview of the process.

In addition to the citation-based measures, for each article, the number of recent tweet posts mentioning it is calculated, as well. This is considered a measure of its social media attention. The *COVID-19-TweetIDs*[5] dataset [28] is used for the collection of COVID-19-relevant tweets. This dataset contains a collection of tweet IDs, each of them published by one of 9 predetermined Twitter accounts (e.g., @WHO) and containing at least one out of predefined coronavirus-related keywords (e.g., "Coronavirus", "covid19", etc). At the time of writing, a subset of this dataset containing tweets posted from (unique tweet IDs) have been integrated in BIP4COVID19. The corresponding Tweet objects were collected using the Twitter API. The result was a collection of tweet objects. The difference between the number of IDs and hydrated objects is due to facts, such as the deletion of tweets in the meantime, which makes some tweets impossible to retrieve.

To find those tweets which are related to the articles in our database, we rely on the URLs of the articles in doi.org, PubMed, and PMC. These URLs are easily produced based

[3]https://github.com/echen102/COVID-19-TweetIDs
[4]https://zenodo.org/record/4774875
[5]https://github.com/echen102/COVID-19-TweetIDs

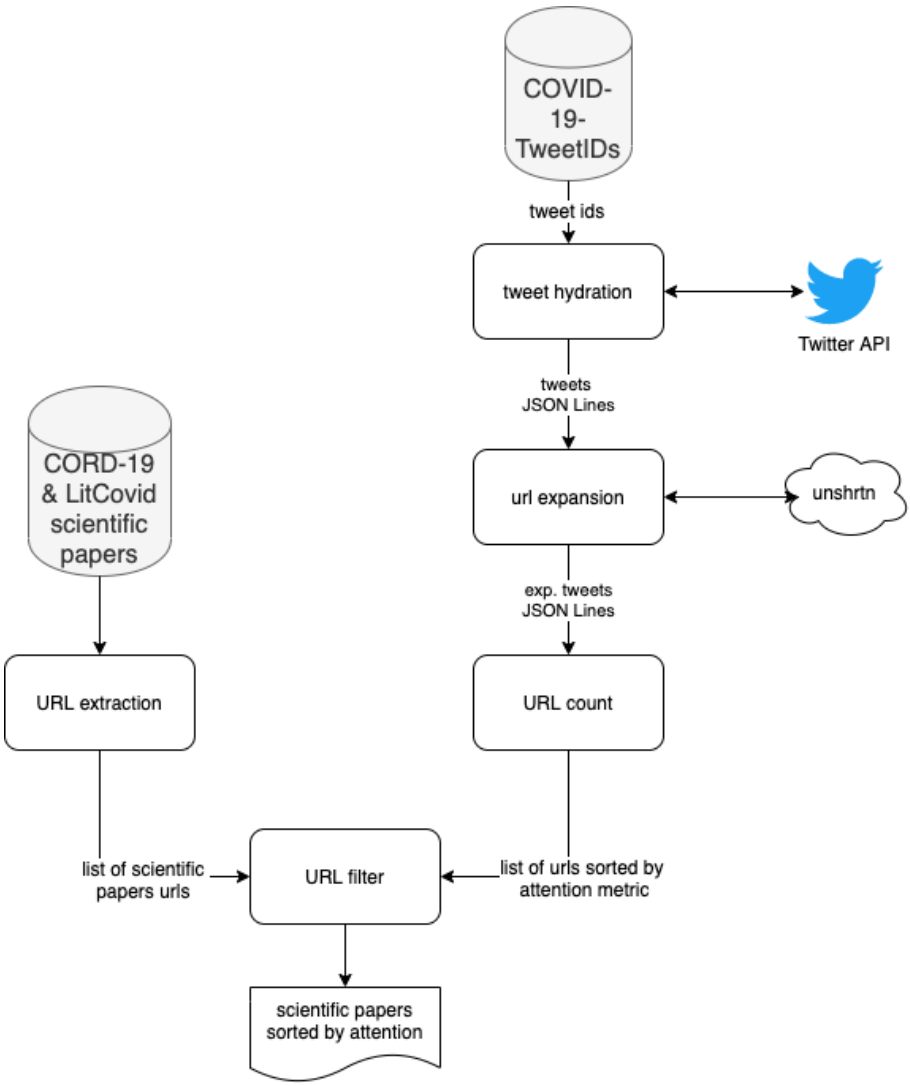


Figure 6.1: Measuring attention on social media for articles related to COVID-19

on the corresponding identifiers. In addition, when possible, the corresponding page in the publisher's website is also retrieved based on the doi.org redirection. After the collection of the URLs of all articles, the number of appearances of the URLs related to each one are produced. However, since the Twitter API returns either shortened or not fully expanded URLs, the fully expanded URLs are collected using the unshrtn[6] library.

6.2 CitySense: Combining Geolocated Data for Urban Area Profiling

Social networks, available open data, and massive online APIs provide vast data about our surrounding location, especially for cities and urban areas. Unfortunately, most previous applications and research usually focused on one kind of data over the other, thus presenting a biased and partial view of each location in question, hence partially negating the benefits of such approaches. To remedy this, we developed the CitySense framework [70] that simultaneously combines data from administrative sources (e.g., public agencies), massive Point of Interest APIs (Google Places, Foursquare), and social microblogs

[6]https://github.com/docnow/unshrtn

(Twitter) to provide a unified view of all available information about an urban area, in an intuitive and easy to use web-application platform. This section describes the engineering and design challenges of such an effort and how these different and divergent sources of information may be combined to provide an accurate and diverse visualization for our use case, the urban area of Chicago, USA.

6.2.1 Introduction and motivation

The emergence of social networks, microblogging platforms, check-in applications, and smartphone / Global Positioning System (GPS) devices in recent years has generated vast amounts of data regarding the location of users. To exploit this vastly growing data, recent research has focused on utilizing the geographic aspect of this information for statistical profiling of geographical areas, event detection, sentiment analysis of users, place-name disambiguation, identification of popular hotspots and their temporal variation, identifying and visualizing the typical movement pattern of users throughout the day, as well as improving existing city maps. However, volunteered geographic information (VGI) contributed by online users is imprecise and inaccurate by design, and it should be used with extra caution for critical applications.

Likewise, the increasing necessity for efficient location-based services and effective online advertising drove leading web providers (e.g., Google, Here, Bing, Foursquare) to store and offer Point of Interest (PoI) information to their users, usually through the use of online Application Programming Interfaces (APIs). Such an approach has several benefits since the users not only have access to information about their nearby PoIs, but they may also provide (or view) reviews or notify their friends of their current whereabouts. The same web services also allow shop-owners and enterprises to advertise their stores and the services they offer. However, as with any commercial offering, there are limitations on using those APIs, thus providing users with a very locally limited view of the existing city infrastructure that cannot be directly used to extract additional information for city-scale areas.

On a separate front, the open data movement argued that citizens should have access to the data collected by government agencies since they are the ones funding data collection through their taxes. A second strong supporting argument is that public access to government data helps individuals and enterprises to create apps that boost the economy and provide better services to the citizens at no additional cost. Some countries and cities have openly released such data, which provides another alternative view of urban areas. Although this open data is official, curated, of excellent quality, and impossible to collect by individuals, it has the obvious disadvantage that it cannot be real-time, it is usually not available through APIs, and most importantly, it may be updated at very infrequent intervals (e.g., census data), therefore at risk of being rather outdated.

Overall, the three sources of information mentioned above, i.e., volunteered geographic information, online PoI data, and official open data, each has its strengths and weaknesses regarding accuracy, update rate, ease of use, and availability. Likewise, applications or research that utilize and rely on only one of those data types offer a biased and imprecise view of reality that could potentially be misleading. To remedy this, this section presents the CitySense framework that utilizes open data from administrative sources, online PoI APIs, and social microblogs (tweets) to provide a unified view of our use case, the urban area of Chicago. The main innovation and focus of the paper is to show how disparate datasets of various origins can be combined to provide a more complete picture of a geographical area. We also present the corresponding web application [7]. Our emphasis is on how to efficiently spatially aggregate, visualize and present the end-user with an

[7] http://geoprofiler.imsi.athenarc.gr/

aesthetically pleasing and intuitive view of available raw data for any of these three sources, with minimum intervention, so that the end-user could freely interpret this information at his own will. As such, the CitySense application could be easily extended with additional features with minimal effort. Moreover, the CitySense Database is designed to store and retrieve the data acquired from the three aforementioned sources.

Overall, CitySense is a dynamic urban area viewer that integrates various datasets related to an urban area, providing a rich visualization of a city's life. As a motivating example, consider a newcomer to the city, who has to search for a house in an unfamiliar area. She has to answer some questions in order to narrow down and locate the neighborhoods to search. These questions may involve criteria like education facilities ("Where are the most popular residential neighborhoods having high-level educational facilities?") and security ("Where is the downtown area with the lowest criminality measures?"). As another example, consider a tour operator that needs to track the tourist activity in a city to offer improved tour packages and services. However, monitoring massive tourist activity using traditional methods would require lots of effort, examining many updating sources, hence huge costs and time involving off-line on-the-spot observation.

The outline of this work is as follows. Section 6.2.2 presents related work. Section 6.2.3 describes the objectives, the architecture, and the web-based application of CitySense. Section 6.2.4 describes the CitySense technical challenges. Section 6.2.5 describes the CitySense Database design. Finally, Section 6.2.6 gives conclusions and directions for future work.

6.2.2 Related work

In recent years, as data from location-sharing systems are constantly increasing, researchers have proposed a wide variety of "urban sensing" methods, based on location data derived from all kinds of sources: social media posts and check-ins, cellphone activity, taxicab records, demographic data, etc. Scientists combined social sciences, computer science, and data mining tools to derive valuable knowledge regarding the lives of cities. Cranshaw et al. [38] tried to reveal the dynamics of a city based on social media activity, while in [149, 81], authors characterized sub-regions of cities by mining significant patterns extracted from geo-tagged tweets. Frias-Martinez et al. [51] focused on deriving land uses and points of interest in a specific urban area based on tweeting patterns, and Noulas et al. [95] analyzed user check-in dynamics to mine meaningful spatio-temporal patterns for urban spaces analysis. Much work has been done using social media textual and semantic content for urban analysis purposes. For example, Pozdnoukhov et al. [105] conducted a real-time spatial analysis of the topical content of streaming tweets.

Moreover, Noulas et al. [96] proposed the comparison of urban neighborhoods by using semantic information attached to places that people check-in, while Kling et al. [76] applied a probabilistic topic model to obtain a decomposition of the stream of digital traces into a set of urban topics related to various activities of the citizens using Foursquare and Twitter data. Grabovitch-Zuyev et al. [56] studied the correlation between textual content and geospatial locations in tweets, and Kamath et al. [67] used the Spatio-temporal propagation of hashtags to characterize locations. Prediction methodologies have widely used geo-tagged social content. For example, Kinsella et al. [75] created language models of locations extracted from geo-tagged Twitter data, in order to predict the location of an individual tweet, in [39, 37, 33, 20], the authors aimed to model friendship between users by analyzing their location trails. Cheng et al. [31] estimated a Twitter user's city-level location based purely on the content of the user's tweets.

Moreover, researchers have focused on trend and event detection by detecting correlations between topics and locations [24, 80]. Lately, many works have been published

focusing on urban mobility patterns. For example, Veloso et al. [142] analyzed the taxi-cab trajectory records in Lisbon to explore the distribution relationship between pick-up locations and drop-off locations. In [82], the authors explored real-time analytical methodologies for spatio-temporal data of citizens' daily travel patterns in an urban environment. The authors of [109, 110, 55, 26, 137] used the moving trajectory data of mobile phone users to study city dynamics and human mobility, while the authors of [34, 138, 61, 32] analyzed human mobility using social media data. Another field connected to urban analysis is the geodemographic classifications, representing small area classifications that provide summary indicators of the social, economic, and demographic characteristics of neighborhoods [60]. In the area of location demographics and socio-economic prediction and correlation, researchers have proposed a variety of methods based on geo-tagged social media data [85, 63, 84]. A wide variety of applications that describe the life of urban areas have been developed so far. For example, EvenTweet [8] is a framework to detect localized events in real-time from a Twitter stream and to track the evolution of such events over time. Moreover, the "One million Tweet Map" [4] is a web app that displays the last million tweets over the world map in real-time. Every second the map is updated, dropping twenty of the earliest tweets and plotting out the latest twenty keeping the number of tweets hovering at 1,000,000, showing clustered tweets in regions around the world, while users can zoom in or out on the map, and cause the re-aggregation of the clusters. Furthermore, the "tweepsmap" [7] application provides users with efficient geo-targeted Twitter analytics and management, and "trendsmap" [6] and "tweetmap" [2] show the geo-located latest trends from Twitter on a map. In Urban Census Demographics visualization field, the "Mapping America: Every City, Every Block" [3] enables users to browse local data from the Census Bureau's American Community Survey, based on samples from 2005 to 2009. Finally, "Social Explorer" [5] provides map-based tools for visual exploration of demographic information, including the U.S. Census, American Community Survey, United Kingdom Census, Canadian Census, Eurostat, FBI Uniformed Crime Report, American election results, Religious Congregation Membership Study, World Development Indicators. Although those works provide thorough insights into some aspects of life in an urban area, they fail to provide an integrated and global view of the city and enable the user to answer questions by combining datasets interactively. CitySense [1] aims to fill these gaps by integrating multiple data sources and providing an interactive user interface supporting filters, multiple view options, and drill-down abilities.

6.2.3 Browsing integrated city data

In this section, we present an overview of CitySense. We also discuss the objectives and present the features of the application.

6.2.3.1 Objectives and Architecture

CitySense is a dynamic urban area viewer that integrates various datasets related to an urban area and provides a rich visualization of a city's life. The application can answer questions at many levels by exploiting the variety of datasets referring to a city and joining disparate data sources in an easy way. Users can view several aspects of city life statically or over time, for the whole city or each part, mixing data sources to uncover patterns and information that would not be obvious from just observing the datasets. The CitySense application [1] aims to provide a fast and easy way to:

- combine disparate data sources regarding various city aspects,
- filter data and drill down through a map-based visualization environment, and

- answer questions, explore, and discover valuable information to convey the sense of the city.

The system architecture is presented in Figure 6.2 and includes the front-end Web-based Application of CitySense, the Data Infrastructure and Refresher units, the GeoServer that is discussed in Section 6.2.4.3, and the CityProfiler subsystem (the dotted box in Figure 6.2) that was developed to collect the data related to the city from the data sources and is presented in detail in Section 6.2.3.2

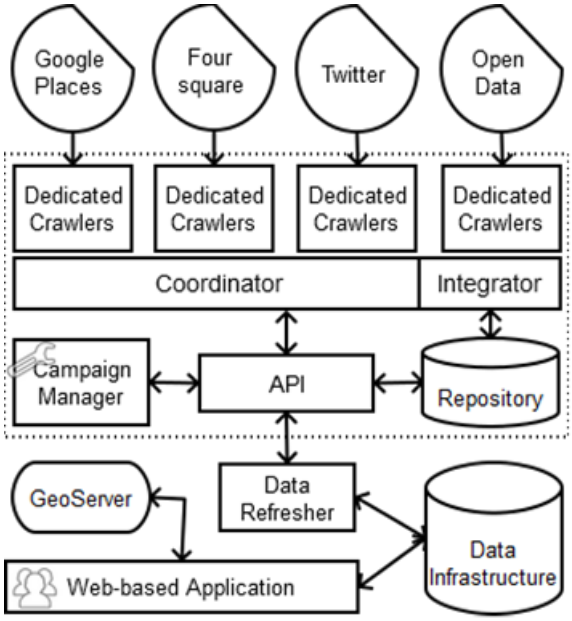


Figure 6.2: CitySense architecture

A screenshot of the CitySense web-based user interface is shown in Figure 6.3. The city of Chicago was selected for the pilot application due to the amount and quality of available official census data. An additional reason is that Chicago's residents are exhibiting strong social media activity; moreover, a sufficient number of Points of Interest (PoIs) is also available.

6.2.3.2 Harvesting Data with CityProfiler

CityProfiler (included in the dotted box in Figure 6.2) is a subsystem of CitySense, responsible for collecting data related to an urban area from diverse sources. Its basic functionality is to collect all available PoIs and tweets from the city and store them in a repository together with relevant metadata.

Specifically, PoI data are extracted using Google Places and Foursquare APIs. Social Media data containing geospatial information are available via the Twitter API. The diversity of these sources raised the need for developing specific modules, called crawlers, to handle each data source. PoI crawlers collect PoI information using two methods. The first (general) method requires selecting a geographic area and a PoI category and returns a list of PoIs in the area belonging to this category. The second (special) method requires selecting a PoI using a unique identifier and returns additional PoI information (name, address, phone number, opening hours, rating, etc.). In any case, the first (general) method returns the unique identifier of each PoI contained in the response list. This identifier can be used by the second (special) method to obtain more information about that PoI. The data obtained by the second method are stored in the repository.

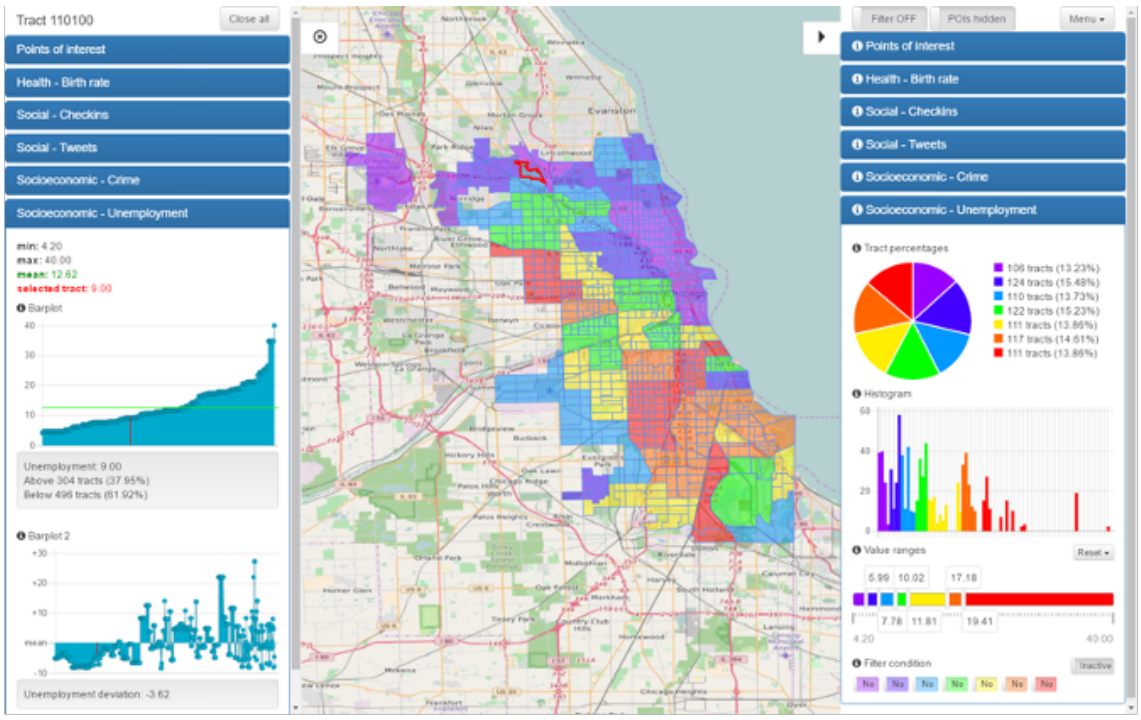


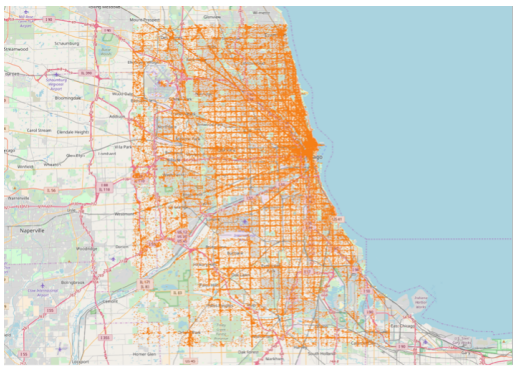
Figure 6.3: CitySense web-based user interface

In the case of the collection of geo-located tweets and check-ins, the corresponding crawler uses a method that requires selecting a geographic area and returns a list of geo-located tweets and check-ins, which have been posted from this area. Specifically, the crawler employs the Twitter Streaming API that provides real-time streaming data. Hence, the crawler has to collect geo-located tweets and check-ins dynamically. This is achieved by using the Twitter Streaming API in a sliding time window. Finally, the data obtained are stored in the repository.

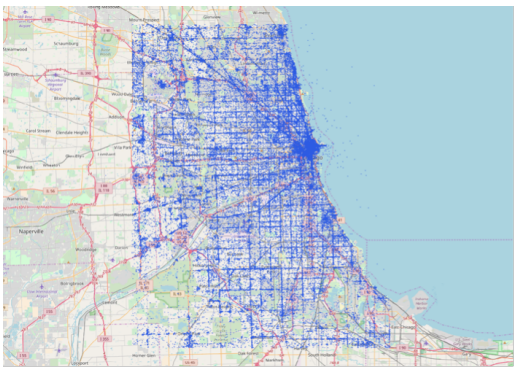
Specifically, Google Places Crawler took 48 hours to complete the Chicago PoI collection. The 184392 PoIs collected and stored in the database are depicted in Figure 6.4a. Meanwhile, the Foursquare Crawler had collected and stored 93893 PoIs that are depicted in Figure 6.4b. Figure 6.4c depicts the complete PoI collection from both Google Places and Foursquare Crawlers. Finally, Figure 6.4d depicts the locations of 10286 geo-located tweets (shown as blue dots) and 1310 check-ins (shown as orange dots) that were collected by Social Media Crawler within these 48 hours.

CityProfiler provides an API and a GUI through which applications and users, respectively, can define and perform new collection campaigns. Each campaign, which is defined by certain parameters, results in an independent collection. These parameters control the individual crawlers that gather data through available APIs and are the following:

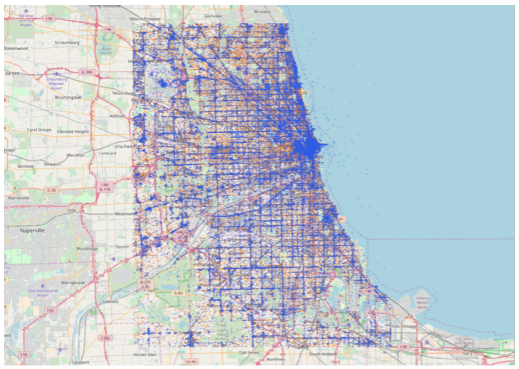
- Crawling Duration: defines the duration of the campaign.
- Crawler Selection: selects which of the available crawlers (corresponding to distinct data sources like Foursquare, Google Maps, Facebook, Twitter, etc.) will participate in the campaign.
- Crawling Location: defines a crawling location by setting a point on the map and a range around it.
- Category Selection: selects target PoI categories and optionally keywords for the crawling to be based on. Keywords are used to narrow crawling when the PoI category employed is deemed too broad (e.g., keyword ”high school” is used when



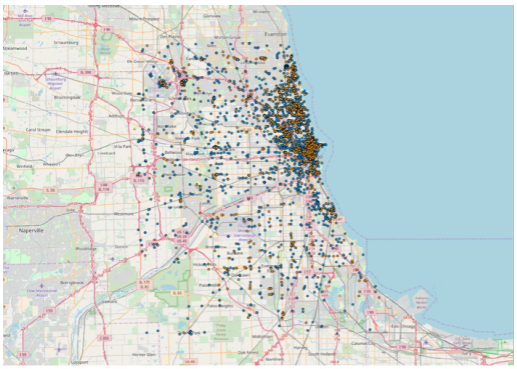
(a) Chicago Google Places PoIs



(b) Chicago Foursquare PoIs



(c) Chicago Google Places and Foursquare PoIs



(d) Geo-located tweets and check-ins locations in Chicago

Figure 6.4: Chicago PoIs

crawling Google Places for high schools, since "school" is the only applicable category). Category Selection can also collect all PoIs in a location, regardless of their category.

- Crawling Frequency Selection: some of the collected data need a systematic update because of the changes that might occur to PoIs (e.g., a coffee shop might become a bar or new PoIs might show up). CityProfiler can perform repetitive campaigns with a large duration in which multiple collections can be performed using the same parameters. Frequency Selection defines, therefore, how often the campaign should automatically restart.

CityProfiler can perform multiple campaigns in parallel. Therefore there is a need for a Coordinator (see Figure 6.2) to control the crawlers and manage the campaigns. Moreover, CityProfiler manages resources in an intelligent way, ensuring that all the restrictions imposed by the sources are met (e.g., the maximum number of requests per time period), and that overlapping requests are avoided. Retrieved data are cleaned to exclude duplicates and are temporarily stored in a repository.

6.2.3.3 Data Preprocessing and Integration

CitySense aims to shed light on the life of a city by exploiting three types of data: Points of Interest, Social Media, and Open Census Data. PoI and Social Media Data are generated constantly by users and services. Therefore, we collect and update them regularly and automatically using CityProfiler, as discussed in Section 6.2.3.2. Unlike these types of data, Open Census Data are generated by diverse sources (local authorities) at unpredictable time intervals. Moreover, they are published in various data formats (CSV,

tab-delimited, etc.). Therefore, Open Census Data require a case-dependent preprocessing and integration procedure keeping pace with their publication and considering the variety of data sources and formats. Finally, the diverse nature of these datasets requires a special integration regarding the aspect of time as well.

An example of the preprocessing and integration transformation regarding Crime data is presented in Figure 6.5. On the left side of the figure, we observe a single row of crime data downloaded in CSV format. This row represents a crime incident and contains its time and location. On the right side of the figure, we observe how this crime is represented in our database. Specifically, it is assigned to a tract (a specific geographical partition of the city) based on its location. The specific crime instance is represented by increasing the counter (total_crimes) in four tables, representing a different time granularity: per year, month, day of the week, and hour of the day.

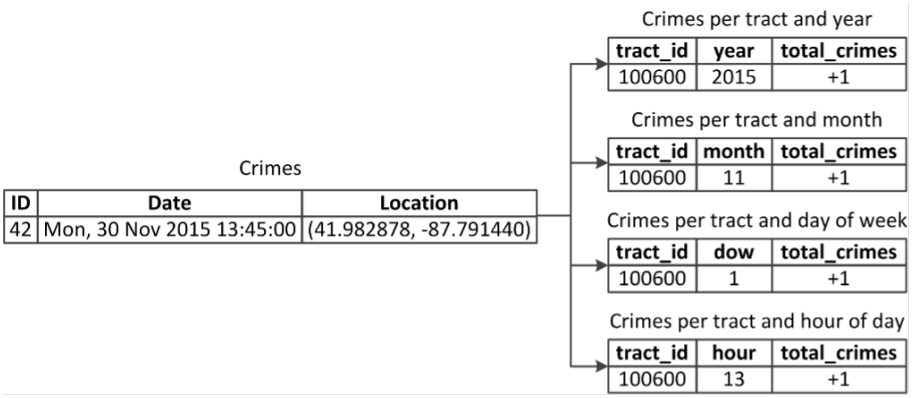


Figure 6.5: Crime data preprocessing and integration example

6.2.3.4 CitySense Features and Design

Figure 6.3 shows CitySense web-based user interface. The central element of the visualization is the map of Chicago, which is divided into smaller sections called tracts. Tracts are existing administrative divisions already used by the Chicago city government departments. Chicago contains 801 tracts, and each of them describes a small area that is considered to be relatively uniform and corresponds ideally to about 1200 households (2000-4000 residents). Tract boundaries are always visible (blue line) on the map, and when an individual tract is chosen, its boundaries are highlighted with a red borderline.

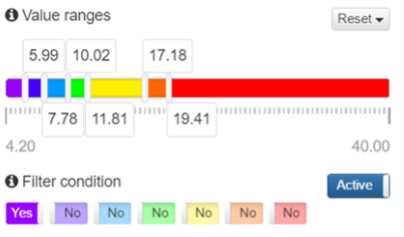
On the two sides of the map, CitySense provides two complementary views of Chicago. The first view appears on the right side and provides functions regarding the city as a whole. Hence, users can define visualization and filter options and observe the results both on the city map coloring and distribution charts. The second view is on the left side and provides charts concerning only the selected tract, dark-highlighted on the map. This view, which appears when a tract is selected, helps users drill down to observe each tract's special characteristics and compare it with the city's overview. These views can be active concurrently, enabling users to observe different datasets at a general level and at tract-level at the same time.

Both views provide visualizations and charts tailored to the corresponding dataset. For example, as shown in Figure 6.3, map coloring and charts visualize the Unemployment dataset.

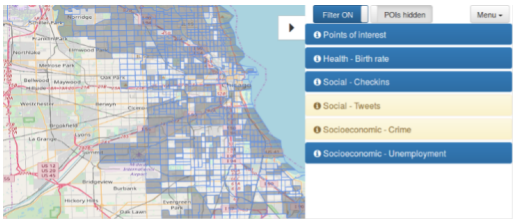
To select a dataset, the user has to select a data drawer. Data drawers (dark rectangles) can be accessed concurrently in both views and represent the available datasets, e.g., "Points of Interest", "Health - Birth rate", "Social – Tweets", etc. According to the type of the particular dataset (see Section 6.2.3.1), each data drawer can contain different UI

elements like pie charts, histograms, color range sliders and implement suitable functionality like a value-based map coloring, temporal and combined filtering, and superimposed PoI information.

The map coloring is based on user adjustable color range sliders that are available in each data drawer. Such a slider is presented in Figure 6.6a (top). After the color ranges are adjusted, users can define one or more colors as filtering parameters for combining various datasets. In other words, CitySense combines datasets (data drawers) by filtering the tracts based on their color. A color filtering slider, where only the violet color (leftmost) is defined as filtering condition, is shown in Figure 6.6a (bottom). The tracts that satisfy the conditions set in all data drawers are colored grey on the map. Figure 6.6b shows the filter output for Social-Tweets and Socioeconomic-Crime datasets. Certain datasets

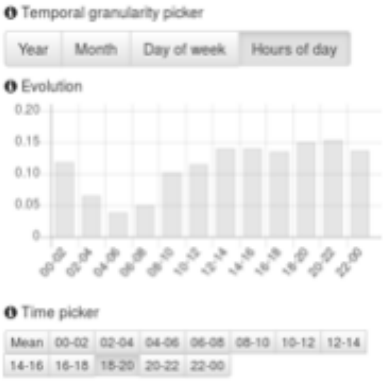


(a) Coloring and filtering color slider

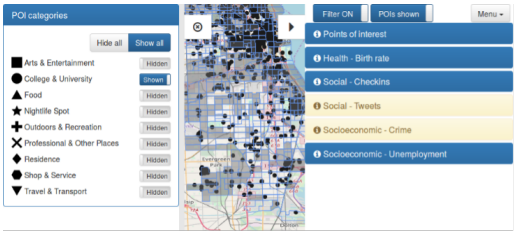


(b) Filtered map

are visualized based on temporal aspects (per month/day/hour). The temporal functions described here are shown in Figure 6.7a. Thus, users can select the time granularity, e.g., the month of the year, day of the week, the hour of the day, to adjust the charts and map coloring accordingly. Additionally, users can color the map or view the tract charts based on a specific month, day, or two-hour interval. Finally, the CitySense application enables the user to see superimposed PoI information on the map at any moment. The user can select one or more categories (Food, Residence, Outdoors & Recreation, etc.), and the corresponding PoIs appear on the map as shown in Figure 6.7b.



(a) Temporal pickers



(b) Filtered map with PoIs

6.2.4 Technical challenges

In this section, we present in detail the technical challenges of the CitySense application.

6.2.4.1 Organizing Disparate Datasets

In order to convey the sense of a city, CitySense must integrate and visualize a variety of datasets. The data sources that are integrated consist of demographic, social media, and

PoI data. The diverse nature of these datasets requires a different integration manipulation regarding the aspect of time. As we show in Table 6.1:

- Open Census Data can be visualized both in a static (overtime) or temporal way (per month/ day/hour). For instance, Health and Unemployment data are visualized statically and Crime data temporally.

- Social Media Data can be visualized in a static, temporal, or dynamic way, although they are produced and gathered dynamically (real-time). The feature of real-time dynamic visualization of social media data is currently being developed.

- Point of Interest Data are visualized in a static way.

	static	**temporal**	**dynamic**
Open Census Data	✓	✓	×
Social Media Data	✓	✓	✓
Point of Interest Data	✓	×	×

Table 6.1: Diversity of dataset visualization regarding time

Diversity of dataset visualization regarding time

The above organization of data helped overcome their diversity and provide coherent visualization and treatment within the application.

A related problem is that of the initialization of the user-adjustable color range sliders. Our goal was to provide a reasonable use of map coloring to help users draw conclusions about the city. Therefore, we provided two options for initialization. The first, the value-based initialization option, breaks the slider based on equidistant values. However, this approach is sensitive to data with extreme outlier values or extreme concentration in certain ranges. The second option provides a percentage-based initialization, hence breaks the slider based on equal distribution percentages. However, this approach is sensitive to having many tracts with almost equal values. As an example, Figure 6.8 shows the value-based initialization for crime data.

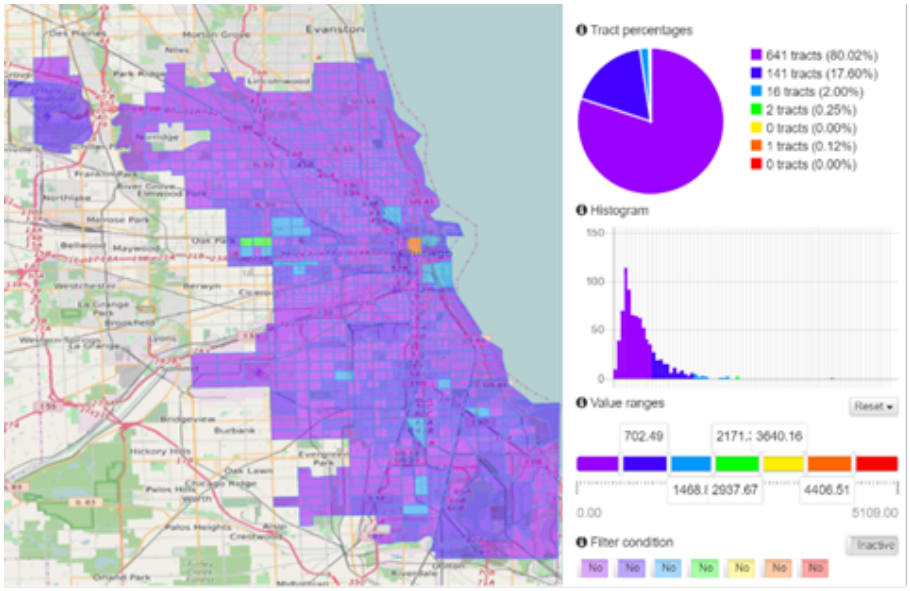


Figure 6.8: Value-based initialization

As we can observe in the histogram shown in Figure 6.8 (right), the crime data mainly occupy a small value range, between 0 and 1468, resulting in the almost two-colored map (violet and indigo – colors may not be visible on printed document) of Figure 6.8 (left). To address this issue, we use the percentage-based initialization, which is presented in Figure 6.9.

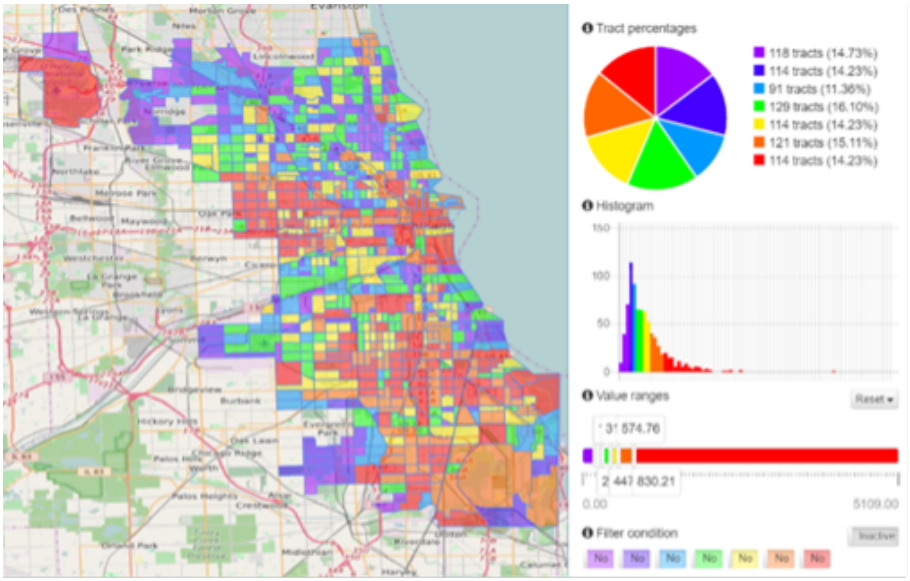


Figure 6.9: Percentage-based initialization

The resulting map coloring shown in Figure 6.9 (left) is obviously improved. However, as we can observe in the tract percentages shown in Figure 6.9 (right), the crime data distributions are not equally divided because some tracts have almost equal values with respect to the range step and, therefore, cannot be equally classified.

6.2.4.2 Acquiring Data of an Area

CityProfiler gathers PoI data from an urban area by performing calls to API services like Google Places and Foursquare, which set restrictions and constraints. A naïve crawling of PoIs, in terms of a whole city, would not be able to collect the entire amount of PoIs, but only a small portion of it as dictated by the rules imposed by the source. CitySense deals with this issue by breaking the area into smaller parts in advance. Specifically, the city is divided into squares of longitude and latitude of 0.03 degrees before the PoI crawling. In case this method does not gather all the PoIs, then recursion is used.

Additionally, CityProfiler collects real-time social data from the city. In order to achieve this, CityProfiler performs a real-time crawling of tweets with Twitter Streaming API, using a location box, which encompasses the city as a filter parameter. Only the geo-located tweets (that are posted along with their latitude and longitude) are collected. In order to collect social check-ins and the PoIs that they were posted at, CityProfiler performs a call to Foursquare API every time a tweet contains a Swarm (mobile app that allows users to share their location within their social network) link. This way, the application collects temporal information concerning geo-located tweets, including their hashtags and check-ins posted at city PoIs.

6.2.4.3 Implementation and Efficiency Issues

Several implementation decisions had to be made so that the application would run efficiently. The application needed to be lightweight with respect to memory and processing

power consumption and responsive concerning the end-user experience.

At certain parts of the application, many geometries, namely tens of thousands of PoIs, need to appear on the screen simultaneously. The option of handling each geometry as a separate entity and drawing it on the map separately would require much memory and processing power, especially when zooming in and out the map. The approach employed is based on drawing relevant geometries as one image layer containing all geometries. GeoServer (shown in Figure 2) is leveraged for generating and serving image layers. For further efficiency, the built-in caching functionality of image layers by GeoServer is utilized. This way, subsequent requests may use already generated image layers.

The application's requirements involve aggregate queries on data, spanning the geospatial and temporal dimensions. Such queries take much time if performed on raw data, resulting in degradation of responsiveness for the end-user. In order to avoid costly operations during runtime, a preprocessing stage is employed. The database design for preprocessed data was driven by the critical use cases available to the end-user via the UI. For example, the user can query for check-in data, aggregated per tract, pertaining to a specific PoI category and a specific day of the week. Raw check-in data contain the geographic coordinates of the PoI, the category of the PoI, and the date and time of the check-in, across two tables. Tract geometries are stored in a separate table as well. Such a query cannot be executed instantaneously. During the preprocessing stage, the coordinates of the PoIs are mapped to the intersecting tracts, the days of week are extracted from date and time, and aggregation per tract and day of week is performed. The preprocessing results are stored in database tables. This way efficient querying for check-ins, in a specific PoI category, on a specific day of week, is achieved. Separate tables are employed to deal with different time granularity aspects of the temporal dimension, i.e., there exist separate tables for years, months, days of the week, hours of the day. Another optimization measure in the same direction is the delegation of heavy computations to the initialization stage of application services. This affects the start-up time of the application but speeds up requests during runtime.

The application currently encompasses a relatively small number of datasets, so data handling is manageable using a PostgreSQL database system, as described in Section **??**. If the datasets grow in number, a data warehouse can be used to facilitate data management and efficient processing of aggregate queries.

6.2.4.4 Adapting to Other Cities

One of our primary concerns during the development of the CitySense framework was the adaptability of the framework to other cities. Adaptation of CitySense to another city comprises three major tasks: partitioning the city area, integrating Open Census Data and implementing the relevant access methods, and specialization of the front-end according to the available city-data.

City Area Partitioning CitySense is essentially parametric with respect to the attributes that define the city of interest, namely a bounding rectangle that encloses the city and a partitioning scheme for the city. In theory, the partitioning scheme may consist of an arbitrary set of polygons that collectively cover the whole city. Choosing a partitioning scheme is, nevertheless, not that straightforward. In order to effectively choose a partitioning scheme, official administrative partitioning schemes should be looked into (e.g., community areas, ZIP codes, census tracts), focusing on partitioning schemes used in Open Census Data of interest. Disregarding such partitioning schemes and employing an arbitrary one could result in Open Census Data of interest rendered useless or hard to map to the employed partitioning scheme. Should the official partitioning scheme be considered too fine-grained, grouping could be applied to the small partitions to acquire a more coarse-grained partitioning scheme to use. Should the official partitioning scheme

be too coarse-grained, segmentation of the large partitions into smaller ones would result in a more fine-grained partitioning scheme to use.

Open Census Data Integration and Access Open Census Data is the most cumbersome type of data to integrate into CitySense. While CityProfiler data are the same, irrespective of the city of interest, Open Census Data could be vastly different, even among different types of Open Census Data for the same city. Open Census Data could be stored in database tables or files. As long as data transfer from the back-end to the front-end is of the same form, all underlying implementation details have no other constraints regardless of the type of data. Open Census Data will often use a specific partitioning scheme that will generally diverge from the partitioning scheme applied to the city. Such data will need to be mapped to the employed partitioning scheme. There is no recipe for universally handling this issue, hence the aforementioned suggestion to let Open Census Data drive the choice of a partitioning scheme for the city. Open Census Data with temporal and/or categorical dimensions should be stored in a way that will facilitate efficient data retrieval based on corresponding parameters. The methods that implement data access should also support temporal and/or categorical parameters, if should such dimensions exist for a specific type of Open Census Data. While parameters are specific to each type of Open Census Data, the response from the back-end should always be of the same form so that all response data can be treated uniformly by the front-end.

Front-end Specialization Specialization of the front-end to support the city data available by the back-end is the final task in the process of CitySense adaptation. Each dataset is represented by a data drawer both in the left and the right sidebar. All datasets follow the same protocol concerning the data sent by the back-end. The only thing that needs to be specialized per dataset is the data picker, in case that one exists for a specific dataset. The data picker is used to navigate categorically and/or temporally within the dataset. The data picker parameters will be transformed to request parameters that are received by the back-end. The back-end response will follow the data transfer protocol. The data drawer, therefore, needs no other specialization before it can display the received data.

Linear Prediction Model Very often, the datasets are not independent of each other. For example, infant mortality is very likely to be income-related and is increased in low-income areas. One way to predict values of a variable (response) based on the corresponding values of other variables (predictors) is to find a suitable linear model based on the method of least squares. There are two reasons for constructing such models:

- They can provide an "exploratory analysis" of data. By comparing the predicted values with the actual, correlations between variables can be explored, e.g., crimes are associated with income and unemployment.
- They can provide an estimation of a missing value for a tract since this value can be inferred based on the values of predictors for this tract.

The CitySense application supports the construction of linear models for any of the available datasets. As an example, we consider crime data. From the application menu, we can create linear models (select "New model fit"), regarding Crime as a response and any combination of predictors. For example, consider Crime as a response, with predictors the Income, Unemployment, Check-ins, and Points of Interest. The result of the model (the prediction for the crime values), which are shown in Figure 6.10, when compared with the actual data for Crime, confirms the association of Crime with the specific predictors. Moreover, before the model construction, there was no crime value for the upper left tract in the dataset. As we observe in Figure 6.10, the same tract has a value and appears in red. Since this tract is selected (red outline), the data for that tract derived from the linear model are shown in the left tray.

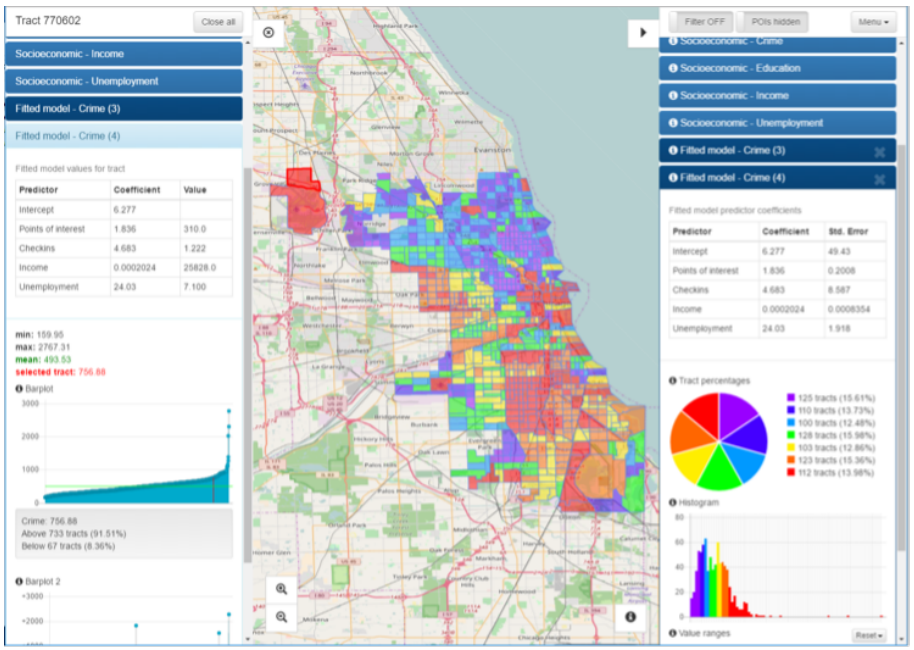


Figure 6.10: **Percentage-based initialization**

6.2.5 CitySense Database

The diverse and complex nature of the data used in the CitySense application poses considerable challenges in data handling and storage. Thus, data come in many different formats and comprise varying content. Therefore, CitySense Database was designed with a flexible structure that can accommodate the diverse styles of the data.

6.2.5.1 Database Requirements

In order to meet the requirements of the application, we opted to use a relational database, which allows the application to retrieve the necessary information based on several criteria immediately. The efficient organization of information regarding PoI, Open Census, and Social Media data requires a database that can adapt to their diverse nature. Specifically:

- Points of Interest require the storage of accompanying features such as name and location. The tract where the PoI is located is also stored to achieve efficient aggregation based on tracts. At the same time, it is necessary to store the PoI category that each PoI belongs to (Arts Entertainment, Food, etc.) so that aggregation based on categories is possible. Finally, Points of Interest do not need separate time information since they are considered static in time.

- Open Census Data comprise both static and temporal data. For example, demographic data, socio-economic indicators, and health indicators are presented as static data and, therefore, temporal information storage is not required for them. On the other hand, crime data require information storage about crime distribution over time (per month, day of the week, etc.).

- Social Media Data, namely tweets and check-ins, are essentially temporal data. It is, therefore, necessary to store the information on their distribution over time. Besides spatial information storage, which is necessary for both tweets and check-ins, our system also needs to store the associated PoI category for check-ins. This is needed for check-in aggregation per PoI category.

Database Design and Schema The main goal of the CitySense project is to exploit as much data as possible for a particular area to have a realistic depiction of its ”trace” on multiple levels. Therefore, the database presented here is designed to meet all the CitySense application requirements and be flexible to adapt to future requirements and store new data that may arise. The corresponding ER diagram of the database is presented in Figure 6.11.

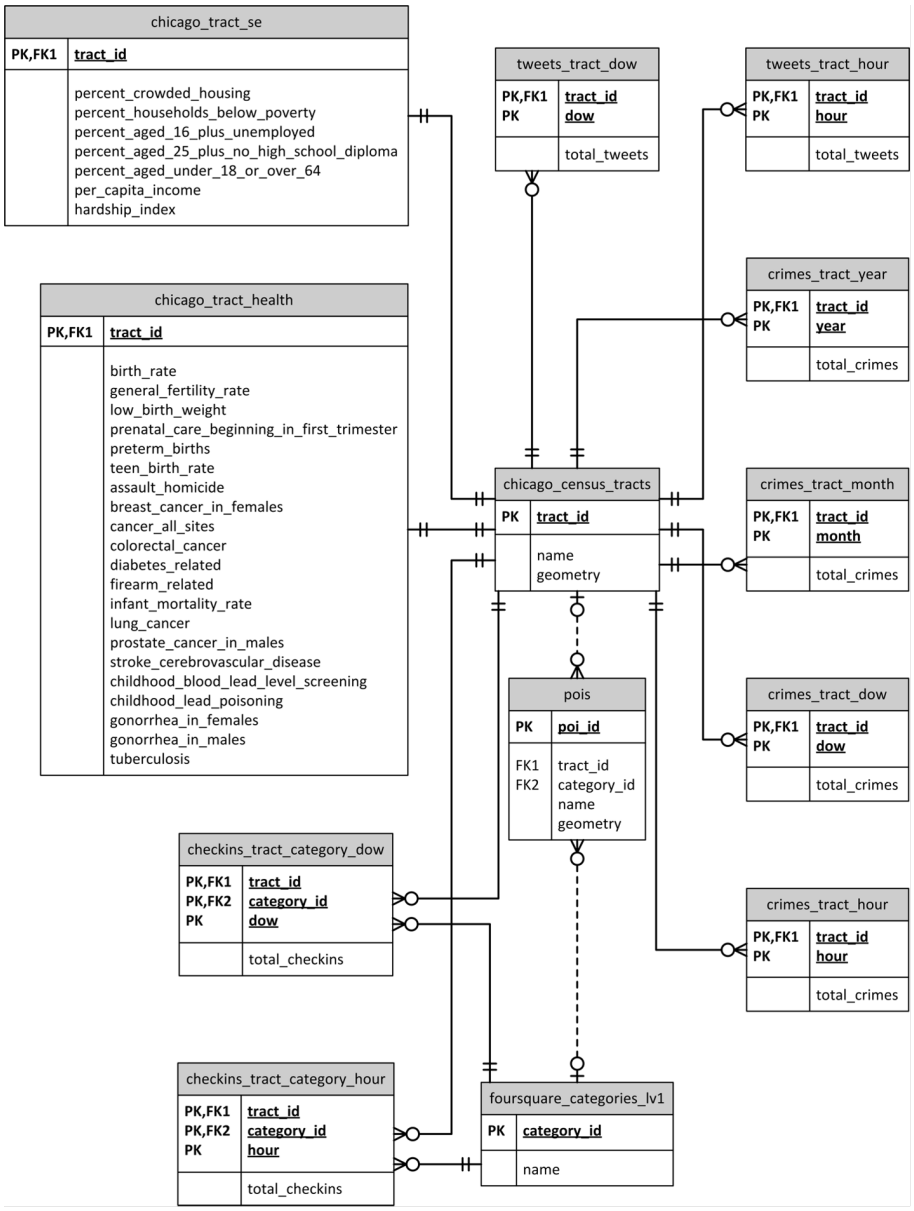


Figure 6.11: **Database schema**

The following is a detailed description of all the tables forming the database:

- **chicago_census_tracts:** The table contains all Chicago tracts that the application supports. It contains the tract's unique identifier, tract name, and polygon geometry that encompasses the tract.

- **chicago_tract_se**: The table is used to store the socio-economic indicator data used by the application. More specifically, it contains the per capita income, poverty rates, unemployment rate, etc., per tract.

- **chicago_tract_health**: The table is used to store the health indicator data used by the application. More specifically, it contains infant mortality indicators, premature births, fertility, etc., per tract.
- **crimes_tract_year**: The table contains the crime data that the application uses. The table stores the number of crimes per tract at a yearly time breakdown.
- **crimes_tract_month**: The table contains the crime data that the application uses. The table stores the number of crimes per tract at a monthly time breakdown.
- **crimes_tract_dow**: The table contains the crime data that the application uses. The table stores the number of crimes per tract and day of the week.
- **crimes_tract_hour**: The table contains the crime data that the application uses. The table stores the number of crimes per tract and time of day.
- **tweets_tract_dow**: The table contains the geo-located tweets data that the application uses. The table stores the number of tweets per tract and day of the week.
- **tweets_tract_hour**: The table contains the geo-located tweets data that the application uses. The table stores the number of tweets per tract and time of day.
- **foursquare_categories_lv1**: The table is used to store the PoI categories that the application uses. The information stored consists of the unique identifier of the PoI category and the category name.
- **pois**: The table is used to store the Points of Interest that the application uses. The information stored consists of the unique identifier of the point of interest, its name, the point geometry that pinpoints the PoI's location, and its category and the tract in which the PoI is located.
- **checkins_tract_category_dow**: The table contains the data of the geo-located tweets that represent check-ins at some point of interest. The table stores the number of check-ins per tract, PoI category, and day of the week.
- **checkins_tract_category_hour**: The table contains the data of the geo-located tweets that represent check-ins at some point of interest. The table stores the number of check-ins per tract, PoI category, and time of day.

The application is using a PostgreSQL database system. PostgreSQL is an open-source relational database management system. To manage spatial information efficiently, we used the PostGIS extension of PostgreSQL, the official spatial extension of PostgreSQL. PostGIS is a software library that adds support for geographic objects (polygons, points) to PostgreSQL databases.

6.2.6 Conclusion

In this section, we presented CitySense, a dynamic urban area viewer that provides a rich visualization of the city's life by integrating disparate datasets. The application helps answer questions and reveals several aspects of city life that would not be obvious from just observing the datasets. In order to accomplish that, we developed special data collection and managing tools, rich visualization and filtering functions, and dealt with several technical challenges. Currently, we are developing the feature of dynamic visualization of social media data (tweet posts, check-ins, and hashtags). The support for dynamic

datasets could be used to cover city power consumption and traffic data in the future. Another future target concerns the incorporation of road network information into our system. Users could calculate the actual distance between PoIs, by exploiting special road network-based functions provided by CitySense. Finally, as more and more data is integrated through CitySense, the problem of scalability will arise. Therefore, a cloud data infrastructure is considered to fit CitySense's future data storing and managing needs.

Chapter 7

Future work

The research interests of this thesis reside at the focus of many research groups worldwide and therefore are continuously advancing. This chapter addresses some open research problems and our main goals for the near future that can be grouped into the following lines of study:

1. **Increasing recommendations quality.** The quality of the recommendations produced needs to be evaluated using criteria that are not limited to prediction accuracy but also include recommendation diversity, serendipity, novelty, and freshness of recommended tweets and followees. Our idea is that the Steiner tree can be further enhanced by employing algorithms and related graph metrics suitable for the new semantic criteria (e.g., graph bridges - betweenness centrality) and time series to capture the interest dynamics.

2. **Integrate additional knowledge graph relationships.** We believe that the quality of the recommendations will improve if our algorithm fully exploits the semantic information contained in richer knowledge graphs (e.g., DBPedia). Therefore, we aim at improving the Steiner tree method so that each type of semantic relation will be treated differently by the algorithm. In this process, the analysis of the importance of semantic relations can play an important role in providing useful weights to the algorithm. Moreover, the relations and features of knowledge graphs can be used to generate transparent and explainable recommendations. Therefore, we aim to focus on using KGs to solve interpretability problems and design interpretable models that lead to the explainability of the recommendation results. It would also be interesting to use KNOwDE's background for allowing users to modify the recommendation criteria and even their profile while browsing the available data (tweets or followees) using the KG.

3. **Fake news detection using the feature-agnostic datasets.** Exploiting the wide range of detection features leads to a critical challenge: minimize the trade-off between classification performance and detection promptness. Therefore, we aim to analyze our feature-agnostic dataset, specifically the user propagation trees, to reveal interesting diffusion patterns and extract new and important detection features. Moreover, we aim to develop new methods for detecting misinformation using both the content and diffusion of news in social media. We aim at using a machine learning (ml) approach in which different sets of features extracted from our datasets will be used to detect the various forms of misleading content. Instead of building a single ml model for fact-checking or intention detection or diffusion process, we will first build a single model for each source of evidence, and then we will blend the individual models by leveraging ensemble learning techniques. We

expect to obtain expressive models for different forms of misleading content that achieve better and more robust overall performance.

4. **Enriching anonymization hierarchies.** Entering the age of digital transformation and big data poses new challenges for protecting sensitive data (health records, medical prescriptions, financial information, online surveys, workplace files, etc.) and complying with privacy rules. K-anonymization refers to protecting private or confidential information by blurring or generalizing identifying data (date of birth, zip code, gender, marital status, etc.) that can be combined to re-identify persons and compromise their privacy. So far, the semantic content of the information has not been used for auto-generating generalization hierarchies. Hierarchy auto-generation for non-numerical data attributes (e.g., profession, city, etc.) results in unclear and incomprehensible hierarchies. Hence, we aim to use knowledge graphs to auto-generate meaningful and comprehensive generalization hierarchies exploiting the semantic relations existing in the graphs.

5. **Enriching knowledge graphs with emerging credible information.** The development of knowledge graphs faces great theoretical and technical difficulties, one of which is recognizing and integrating new knowledge created dynamically in real-time. Most knowledge graph implementations use some existing knowledge bases and hierarchies (Wikipedia, WordNet, Wikidata), which are updated manually. This creates delays in integrating new knowledge resulting in outdated available information. Therefore, a possible solution is to enrich knowledge graphs with content (events, persons, entities, emerging news, and relations) extracted from streaming social media posts. Hence, we plan the development of effective social media data analysis methods to discover new knowledge for enriching knowledge graphs, using event detection, relations extraction, and item/event/node summarization techniques.

Chapter 8

Conclusion

In this thesis, we explored the possibilities of connecting fascinating and up-to-date research areas. Closing this thesis, we would like to highlight the most important contributions of this work. First, we managed to improve the efficiency of social network recommendation systems by utilizing semantic information from knowledge graphs. In addition, we used knowledge graphs to automate the creation of training datasets for the development of efficient misinformation detection models. Finally, we developed methods for exploring and understanding increasingly large and complex datasets by analyzing data from social networks and knowledge graphs.

Specifically, we showed that exploiting the semantic relations between users' topics of interest improves content-based recommenders' efficiency in Twitter. Our method benefits from the objective relations between the topics extracted from knowledge graphs. We also showed that the recommendations based on these relations reduce the effects of over-recommendation, over-specialization, and the cold-start problems and overcome the limitations posed by commercial APIs.

Moreover, we explored the range of the misinformation and disinformation detection features encountered in the literature. This thesis provides a complete feature typology based on the analysis and systematization of all available features. This typology can be useful for training fake news detection models that cover all types of misinformation. We also described PHONY, an infrastructure for automating the generation of feature-agnostic datasets. These datasets contain fake news and their propagation footprints in the Twitter network based on the fake news stories provided by curated fact-checking websites and comprise the necessary data to extract all features encountered in the feature typology.

Furthermore, in the front of data exploration, we present three applications that tackle the problem in the light of knowledge recommendations, social media attention, and data insights visualization. Specifically, we showed KNOwDE that generated keyword recommendations based on knowledge bases, user interaction, and data graph relations that connect the dataset objects. We also present the methodology for calculating an attention altmetric indicator extracted from social media data for ranking literature related to COVID-19 comprised in the Bip4COVID dataset. Finally, we showcased CitySense, a dynamic urban area viewer, and described the underlying framework that combines data from administrative sources, Point of Interest APIs, and social media to provide a unified view of all available information about an urban area.

Bibliography

[1] Citysense. http://citysense.imis.athena-innovation.gr:8080/citysense/. Accessed: 2017-04-08.

[2] Mapd tweetmap. https://www.mapd.com/demos/tweetmap. Accessed: 2017-04-08.

[3] Mapping america: Every city, every block. http://www.nytimes.com/projects/census/2010/explorer.html. Accessed: 2017-04-08.

[4] The one million tweet map. http://onemilliontweetmap.com. Accessed: 2017-04-08.

[5] Social explorer. https://www.socialexplorer.com/explore/maps. Accessed: 2017-04-08.

[6] Trendsmap realtime local twitter trends. http://trendsmap.com. Accessed: 2017-04-08.

[7] Tweepsmap. https://tweepsmap.com. Accessed: 2017-04-08.

[8] H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6(12):1326–1329, 2013.

[9] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[10] S. Afroz, M. Brennan, and R. Greenstadt. Detecting Hoaxes, Frauds, and Deception in Writing Style Online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475, San Francisco, CA, USA, may 2012. IEEE.

[11] N. Aggarwal and P. Buitelaar. Query expansion using wikipedia and dbpedia. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[12] B. Al-Shboul and S.-H. Myaeng. Wikipedia-based query phrase expansion in patent class search. *Information retrieval*, 17(5-6):430–451, 2014.

[13] M. ALMasri, C. Berrut, and J.-P. Chevallet. Wikipedia-based semantic query enrichment. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval*, pages 5–8, 2013.

[14] O. Alonso, C. Carson, D. Gerster, X. Ji, and S. U. Nabar. Detecting uninteresting content in text streams. In *SIGIR Crowdsourcing for Search Evaluation Workshop*, 2010.

[15] M. Amoruso and D. Anello. Contrasting the Spread of Misinformation in Online Social Networks. *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, page 9, 2017.

[16] R. Anand and A. Kotov. An empirical comparison of statistical term association graphs with dbpedia and conceptnet for query expansion. In *Proceedings of the 7th forum for information retrieval evaluation*, pages 27–30, 2015.

[17] Apache. Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene, 2020.

[18] J. Arguello, J. L. Elsas, C. Yoo, J. Callan, and J. G. Carbonell. Document and query expansion models for blog distillation. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST, 2008.

[19] M. G. Armentano, D. Godoy, and A. Amandi. Topology-based recommendation of users in micro-blogging communities. *Journal of Computer Science and Technology*, 27(3):624–634, 2012.

[20] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70, 2010.

[21] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[22] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi. Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 357–360, 2014.

[23] A. Brasoveanu and R. Andonie. Semantic Fake News Detection: A Machine Learning Perspective. In *IWANN - Proceedings International Work-Conference on Artificial Neural Networks*, 2019.

[24] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi. Geoscope: Online detection of geo-correlated information trends in social networks. *Proceedings of the VLDB Endowment*, 7(4):229–240, 2013.

[25] C. Cadwalladr and E. Graham-Harrison. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The Guardian*.

[26] F. Calabrese, M. Diao, G. Di Lorenzo, J. Ferreira Jr, and C. Ratti. Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. *Transportation research part C: emerging technologies*, 26:301–313, 2013.

[27] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web - WWW '11*, page 675, Hyderabad, India, 2011. ACM Press.

[28] E. Chen, K. Lerman, and E. Ferrara. Covid-19: The first public coronavirus twitter dataset. arxiv 2020. *arXiv preprint arXiv:2003.07372*.

[29] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 661–670, 2012.

[30] Y. Chen, N. J. Conroy, and V. L. Rubin. Misleading Online Content: Recognizing Clickbait as ?False News? In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection - WMDD '15*, pages 15–19, Seattle, Washington, USA, 2015. ACM Press.

[31] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768, 2010.

[32] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

[33] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.

[34] A. Chua, E. Marcheggiani, L. Servillo, and A. V. Moere. Flowsampler: visual analysis of urban flows in geolocated social media data. In *International Conference on Social Informatics*, pages 5–17. Springer, 2014.

[35] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. Computational fact checking from knowledge networks. *PLoS ONE*, 10(6):e0128193, jun 2015.

[36] N. J. Conroy, V. L. Rubin, and Y. Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, jan 2015.

[37] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441, 2010.

[38] J. Cranshaw, R. Schwartz, J. Hong, and N. Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 6, 2012.

[39] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128, 2010.

[40] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374, 2014.

[41] E. Diaz-Aviles, L. Drumond, Z. Gantner, L. Schmidt-Thieme, and W. Nejdl. What is happening right now... that interests me? online topic discovery and recommendation in twitter. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1592–1596, 2012.

[42] W. Dong, W. Zhang, and C. W. Tan. Rooting out the rumor culprit from suspects. In *2013 IEEE International Symposium on Information Theory*, pages 2671–2675, Istanbul, Turkey, jul 2013. IEEE.

[43] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H. Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 295–303, 2010.

[44] L. Ehrlinger and W. Wöß. Towards a definition of knowledge graphs. In *SEMANTiCS*, 2016.

[45] L. Ehrlinger and W. Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48:1–4, 2016.

[46] H. G. Elmongui, R. Mansour, H. Morsy, S. Khater, A. El-Sharkasy, and R. Ibrahim. Trupi: Twitter recommendation based on users' personal interests. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 272–284. Springer, 2015.

[47] G. Fanti, P. Kairouz, S. Oh, and P. Viswanath. Spy vs. Spy: Rumor Source Obfuscation. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems - SIGMETRICS '15*, pages 271–284, Portland, Oregon, USA, 2015. ACM Press.

[48] M. Farajtabar, M. Gomez-Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song. Back to the Past: Source Identification in Diffusion Networks from Partially Observed Cascades. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, jan 2015.

[49] S. Feng, R. Banerjee, and Y. Choi. Syntactic stylometry for deception detection. In *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*, volume 2, pages 171–175, 2012.

[50] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[51] V. Frias-Martinez, V. Soto, H. Hohwald, and E. Frias-Martinez. Characterizing urban landscapes using geolocated tweets. In *2012 International conference on privacy, security, risk and trust and 2012 international confernece on social computing*, pages 239–248. IEEE, 2012.

[52] E. Gabrilovich, S. Markovitch, et al. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

[53] R. Ghosh, T.-T. Kuo, C.-N. Hsu, S.-D. Lin, and K. Lerman. Time-aware ranking in dynamic citation networks. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 373–380. IEEE, 2011.

[54] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

[55] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779–782, 2008.

[56] I. Grabovitch-Zuyev, Y. Kanza, E. Kravi, and B. Pat. On the correlation between textual content and geospatial locations in microblogs. In *Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data*, pages 1–6, 2014.

[57] A. Guille and H. Hacid. A predictive model for the temporal dynamics of information diffusion in online social networks. In *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*, pages 1145–1152, 2012.

[58] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. TweetCred: A Real-time Web-based System for Assessing Credibility of Content on Twitter. *Proceedings of the 21st International Conference on World Wide Web*, abs/1405.5, 2014.

[59] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206, 2010.

[60] R. Harris, P. Sleight, and R. Webber. *Geodemographics, GIS and neighbourhood targeting*, volume 8. John Wiley & Sons, 2005.

[61] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti. Geo-located twitter as proxy for global mobility patterns. *Cartography and Geographic Information Science*, 41(3):260–271, 2014.

[62] L. Hong, A. S. Doumith, and B. D. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 557–566, 2013.

[63] D. Hristova, M. J. Williams, M. Musolesi, P. Panzarasa, and C. Mascolo. Measuring urban social diversity using interconnected geo-social networks. In *Proceedings of the 25th international conference on world wide web*, pages 21–30, 2016.

[64] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 1–6, 2010.

[65] A. Jain, V. Borkar, and D. Garg. Fast rumor source identification via random walks. *Social Network Analysis and Mining*, 6(1), dec 2016.

[66] Z. Jin, J. Cao, Y. Zhang, and J. Luo. News Verification by Exploiting Conflicting Social Viewpoints in Microblogs. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 7, 2016.

[67] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng. Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 667–678, 2013.

[68] I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, and Y. Vassiliou. Ranking papers by their short-term scientific impact. *arXiv preprint arXiv:2006.00951*, 2020.

[69] D. P. Karidi. From user graph to topics graph: Towards twitter followee recommendation based on knowledge graphs. In *2016 IEEE 32nd International Conference on Data Engineering Workshops (ICDEW)*, pages 121–123. IEEE, 2016.

[70] D. P. Karidi, H. Nakos, A. Efentakis, and Y. Stavrakas. Citysense: Retrieving, visualizing and combining datasets on urban areas. *DBKDA 2017*, page 70, 2017.

[71] D. P. Karidi, Y. Stavrakas, and Y. Vassiliou. A personalized tweet recommendation approach based on concept graphs. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 253–260. IEEE, 2016.

[72] D. P. Karidi, Y. Stavrakas, and Y. Vassiliou. Tweet and followee personalized recommendations based on knowledge graphs. *Journal of Ambient Intelligence and Humanized Computing*, 9(6):2035–2049, 2018.

[73] N. Kawamae. Trend analysis model: trend consists of temporal words, topics, and timestamps. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 317–326, 2011.

[74] Y. Kim and K. Shim. Twitobi: A recommendation system for twitter using probabilistic modeling. In *2011 IEEE 11th International Conference on Data Mining*, pages 340–349. IEEE, 2011.

[75] S. Kinsella, V. Murdock, and N. O'Hare. " i'm eating a sandwich in glasgow" modeling locations with tweets. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 61–68, 2011.

[76] F. Kling and A. Pozdnoukhov. When a city tells a story: urban topic analysis. In *Proceedings of the 20th international conference on advances in geographic information systems*, pages 482–485, 2012.

[77] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[78] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang. Prominent features of rumor propagation in online social media. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 1103–1108, dec 2013.

[79] H. W. Lauw, A. Ntoulas, and K. Kenthapadi. Estimating the quality of postings in the real-time web. 2010.

[80] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*, pages 1–10, 2010.

[81] R. Lee, S. Wakamiya, and K. Sumiya. Urban area characterization based on crowd behavioral lifelogs over twitter. *Personal and ubiquitous computing*, 17(4):605–620, 2013.

[82] L. Liu, A. Biderman, and C. Ratti. Urban mobility landscape: Real time monitoring of urban mobility patterns. In *Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management*, pages 1–16. Citeseer, 2009.

[83] Y. Liu, X. Chen, S. Li, and L. Wang. A user adaptive model for followee recommendation on twitter. In *Natural Language Understanding and Intelligent Applications*, pages 425–436. Springer, 2016.

[84] A. Llorente, M. Garcia-Herranz, M. Cebrian, and E. Moro. Social media fingerprints of unemployment. *PloS one*, 10(5):e0128692, 2015.

[85] P. A. Longley and M. Adnan. Geo-temporal twitter demographics. *International Journal of Geographical Information Science*, 30(2):369–389, 2016.

[86] C. Lu, W. Lam, and Y. Zhang. Twitter user modeling and tweets recommendation based on wikipedia concept graph. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[87] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. F. Wong, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2016-January, pages 3818–3824, 2016.

[88] J. Ma, W. Gao, Z. Wei, Y. Lu, and K. F. Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the International Conference on Information and Knowledge Management*, volume 19-23-Oct-, pages 1751–1754. ACM, oct 2015.

[89] A. Magdy and N. Wanas. Web-based statistical fact checking of textual documents. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents - SMUC '10*, page 103, Toronto, ON, Canada, 2010. ACM Press.

[90] P. A. Ménard and A. Mougeot. Turning silver into gold: Error-focused corpus reannotation with active learning. In *International Conference Recent Advances in Natural Language Processing, RANLP*, volume 2019-September, pages 758–767, 2019.

[91] T. Mitra and E. Gilbert. CREDBANK: A Large-Scale Social Media Corpus with Associated Credibility Annotations. *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM*, page 10, 2015.

[92] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd international web science conference*, pages 1–7, 2011.

[93] D. T. Nguyen, N. P. Nguyen, and M. T. Thai. Sources of misinformation in Online Social Networks: Who to suspect? In *MILCOM 2012 - 2012 IEEE Military Communications Conference*, pages 1–6, Orlando, FL, USA, oct 2012. IEEE.

[94] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104:11–33, 2016.

[95] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

[96] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.

[97] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[98] J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu. Content based fake news detection using knowledge graphs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11136 LNCS, pages 669–683, 2018.

[99] S. Park, J. Y. Park, J.-h. Kang, and M. Cha. The presence of unexpected biases in online fact-checking. *The Harvard Kennedy School Misinformation Review*, 2021.

[100] M. J. Pazzani, J. Muramatsu, D. Billsus, et al. Syskill & webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.

[101] M. Pennacchiotti, F. Silvestri, H. Vahabi, and R. Venturini. Making your interests follow you on twitter. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 165–174, 2012.

[102] D. Pla Karidi, H. Nakos, and Y. Stavrakas. Automatic ground truth dataset creation for fake news detection in social media. In *Intelligent Data Engineering and Automated Learning – IDEAL 2019 Lecture Notes in Computer Science*, volume 11871 LNCS, pages 424–436, 2019.

[103] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum. Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media. In *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*, pages 1003–1012, Perth, Australia, 2017. ACM Press.

[104] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein. A Stylometric Inquiry into Hyperpartisan and Fake News. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240, Melbourne, Australia, 2018. Association for Computational Linguistics.

[105] A. Pozdnoukhov and C. Kaiser. Space-time dynamics of topics in streaming text. In *Proceedings of the 3rd ACM SIGSPATIAL international workshop on location-based social networks*, pages 1–8, 2011.

[106] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor Has It: Identifying Misinformation in Microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1589–1599, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[107] W. Quattrociocchi, A. Scala, and C. R. Sunstein. Echo Chambers on Facebook. *SSRN Electronic Journal*, 2016.

[108] D. Ramage, S. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[109] C. Ratti, D. Frenchman, R. M. Pulselli, and S. Williams. Mobile landscapes: using location data from cell phones for urban analysis. *Environment and planning B: Planning and design*, 33(5):727–748, 2006.

[110] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive computing*, 6(3):30–38, 2007.

[111] S. Rendle and L. Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 251–258, 2008.

[112] F. M. Rodríguez, L. M. Torres, and S. E. Garza. Followee recommendation in twitter using fuzzy link prediction. *Expert systems*, 33(4):349–361, 2016.

[113] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33. Springer, 2011.

[114] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics.

[115] V. L. Rubin and T. Lukoianova. Truth and deception at the rhetorical structure level: Truth and Deception at the Rhetorical Structure Level. *Journal of the Association for Information Science and Technology*, 66(5):905–917, may 2015.

[116] N. Ruchansky, S. Seo, and Y. Liu. CSI: A Hybrid Deep Model for Fake News Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*, pages 797–806, 2017.

[117] G. C. Santia and J. R. Williams. BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM*, 2018.

[118] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[119] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324. Citeseer, 2002.

[120] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[121] D. Shah and T. Zaman. Rumors in a Network: Who's the Culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, aug 2011.

[122] D. Shah and T. Zaman. Rumor centrality: A universal source detector. *SIGMETRICS Performance Evaluation Review*, 40(1 SPEC. ISS.):199–210, 2012.

[123] D. Shah and T. Zaman. Finding Rumor Sources on Random Trees. *Operations Research*, 64(3):736–755, jun 2016.

[124] C. Shao, G. L. Ciampaglia, O. Varol, K. Yang, A. Flammini, and F. Menczer. The spread of low-credibility content by social bots. *Nature Communications*, 9(1):4787, dec 2018.

[125] A. Sharma, J. Jiang, P. Bommannavar, B. Larson, and J. Lin. Graphjet: Real-time content recommendations at twitter. *Proceedings of the VLDB Endowment*, 9(13):1281–1292, 2016.

[126] B. Shi and T. Weninger. Fact Checking in Heterogeneous Information Networks. In *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, pages 101–102, Montréal, Québec, Canada, 2016. ACM Press.

[127] Y. Shi, M. Larson, and A. Hanjalic. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 125–132, 2009.

[128] J. Shin, L. Jian, K. Driscoll, and F. Bar. Political rumoring on Twitter during the 2012 US presidential election: Rumor diffusion and correction. *New Media and Society*, 19(8):1214–1235, mar 2017.

[129] K. Shu, H. R. Bernard, and H. Liu. Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pages 43–65. Springer, 2019.

[130] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data*, 8(3):171–188, jun 2020.

[131] E. Snowden. *Permanent Record*. Metropolitan Books/Henry Holt, 2019.

[132] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[133] J. Subercaze, C. Gravier, and F. Laforest. Real-time, scalable, content-based twitter users recommendation. In *Web Intelligence*, volume 14, pages 17–29. IOS Press, 2016.

[134] E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro. Some Like it Hoax: Automated Fake News Detection in Social Networks, 2017.

[135] M. S. Tajbakhsh and J. Bagherzadeh. Microblogging hash tag recommendation system based on semantic tf-idf: Twitter use case. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 252–257. IEEE, 2016.

[136] E. C. Tandoc, Z. W. Lim, and R. Ling. Defining "Fake News": A typology of scholarly definitions. *Digital Journalism*, 6(2):137–153, feb 2018.

[137] D. Taniar and J. Goh. On mining movement pattern from mobile users. *International Journal of Distributed Sensor Networks*, 3(1):69–86, 2007.

[138] J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M. C. González. Coupling human mobility and social ties. *Journal of The Royal Society Interface*, 12(105):20141128, 2015.

[139] A. S. A. P. L. Ungar and D. Pennock. Methods and metrics for cold-start recommendations. In *Proc. 25th Ann. Int'l ACM SIGIR Conf*, volume 10, 2002.

[140] I. Uysal and W. B. Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2261–2264, 2011.

[141] C. J. Vargo, L. Guo, and M. A. Amazeen. The agenda-setting power of fake news: A big data analysis of the online media landscape from 2014 to 2016. *New Media and Society*, 20:2028–2049, 2018.

[142] M. Veloso, S. Phithakkitnukoon, and C. Bento. Sensing urban mobility with taxi flow. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 41–44, 2011.

[143] T. Vergoulis, I. Kanellos, S. Chatzopoulos, D. P. Karidi, and T. Dalamagas. BIP4COVID19: Impact metrics and indicators for coronavirus related publications, Oct. 2021. Please cite: Thanasis Vergoulis, Ilias Kanellos, Serafeim Chatzopoulos, Danae Pla Karidi, Theodore Dalamagas. "BIP4COVID19: Releasing impact measures for articles relevant to COVID-19". bioRxiv 2020.04.11.037093; doi: https://doi.org/10.1101/2020.04.11.037093.

[144] T. Vergoulis, I. Kanellos, S. Chatzopoulos, D. P. Karidi, and T. Dalamagas. BIP4COVID19: Releasing impact measures for articles relevant to COVID-19. *Quantitative Science Studies*, pages 1–33, 11 2021.

[145] T. Vergoulis, I. Kanellos, S. Chatzopoulos, D. Pla Karidi, and T. Dalamagas. Bip4covid19: Releasing impact measures for articles relevant to covid-19. *bioRxiv*, 2020.

[146] M. D. Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences of the United States of America*, 113(3):554–559, jan 2016.

[147] S. Vosoughi. *Automatic detection and verification of rumors on Twitter*. PhD thesis, Massachusetts Institute of Technology, 2015.

[148] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, mar 2018.

[149] S. Wakamiya, R. Lee, and K. Sumiya. Crowd-sourced urban life monitoring: urban area characterization based crowd behavioral patterns from twitter. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, pages 1–9, 2012.

[150] M. Wang and J. Ma. A novel recommendation approach based on users' weighted trust relations and the rating similarities. *Soft Computing*, 20(10):3981–3990, 2016.

[151] W. Y. Wang. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection, 2017.

[152] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, 2006.

[153] Z. Wang, W. Dong, W. Zhang, and C. W. Tan. Rumor source detection with multiple observations: fundamental limits and algorithms. In *The 2014 ACM international conference on Measurement and modeling of computer systems - SIGMETRICS '14*, pages 1–13, Austin, Texas, USA, 2014. ACM Press.

[154] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270, 2010.

[155] K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on Sina Weibo by propagation structures. In *Proceedings - International Conference on Data Engineering*, volume 2015-May, pages 651–662, apr 2015.

[156] L. Wu and H. Liu. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, volume 2018-Febuary of *WSDM '18*, pages 637–645, New York, NY, USA, 2018. ACM.

[157] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu. Toward computational fact-checking. *Proceedings of the VLDB Endowment*, 7(7):589–600, mar 2014.

[158] C. Xiong and J. Callan. Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120, 2015.

[159] W. Xu and H. Chen. Scalable Rumor Source Detection under Independent Cascade Model in Online Social Networks. In *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 236–242, Shenzhen, China, dec 2015. IEEE.

[160] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, 2009.

[161] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D. Ragan, S. Ji, and X. B. Hu. XFake: Explainable Fake News Detector with Visualizations. *The World Wide Web Conference on - WWW '19*, abs/1907.0:3600–3604, 2019.

[162] F. Yang, X. Yu, Y. Liu, and M. Yang. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1–7, Beijing, China, 2012. ACM Press.

[163] M.-C. Yang, J.-T. Lee, S.-W. Lee, and H.-C. Rim. Finding interesting posts in twitter based on retweet graph analysis. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1073–1074, 2012.

[164] M.-C. Yang and H.-C. Rim. Identifying interesting twitter contents using topical analysis. *Expert Systems with Applications*, 41(9):4330–4336, 2014.

[165] M. Yigit, B. E. Bilgin, and A. Karahoca. Extended topology based recommendation system for unidirectional social networks. *Expert Systems with Applications*, 42(7):3653–3661, 2015.

[166] L. Zeng, K. Starbird, and E. S. Spiro. #Unconfirmed: Classifying Rumor Stance in Crisis-Related Social Media Messages. In *Proceedings of the International Conference on Web and Social Media*, 2016.

[167] F. Zhao, Y. Zhu, H. Jin, and L. T. Yang. A personalized hashtag recommendation approach using lda-based topic model in microblog environment. *Future Generation Computer Systems*, 65:196–206, 2016.

[168] Z. Zhao, P. Resnick, and Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405. International World Wide Web Conferences Steering Committee, may 2015.

[169] D. Zhou, S. Lawless, X. Wu, W. Zhao, and J. Liu. Enhanced personalized search using social data. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 700–710, 2016.

[170] K. Zhu and L. Ying. Information Source Detection in the SIR Model: A Sample-Path-Based Approach. *IEEE/ACM Transactions on Networking*, 24(1):408–421, feb 2016.

[171] A. Zubiaga, M. Liakata, and R. Procter. Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media. *arXiv:1610.07363 [cs]*, oct 2016.

[172] S. Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. 1st edition, 2018.

Appendix A

Feature typology for misinformation and disinformation detection

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F1	Ratio of tweets containing negations	Content	Text Sentiment Analysis	ratio	Measure each and aggregate by rumor/topic/timespan of event	YES	$\mathrm{NR}(R) = \frac{N(A)}{A}\pi r^2$
varF1	negation words (liwc)	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF1	negate-sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	no, not never
varF1	# of negative words in microblogs	Content	Text Sentiment Analysis	cnt	Measure each and aggregate by timespan of event	YES	number of negative words in the text
varF1	% of negative microblogs	Content	Text Sentiment Analysis	fraction	Measure each and aggregate by timespan of event	YES	% negative words in the text
varF1	sentiment negative words	Content	Text Sentiment Analysis	cnt	Measure each and aggregate by rumor/topic	NO	number of negative words in the text
varF1	presence of negative emotion words	Content	Text Sentiment Analysis	boolean	Each tweet	NO	True/false

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF1	fraction sentiment negative	Content	Text Sentiment Analysis	fraction	Measure each and aggregate by rumor/topic	NO	The fraction of tweets with a negative score
F2	ratio of tweets containing opinion & insight	Content	Text Sentiment Analysis	ratio	Measure each and aggregate by rumor/topic	YES	
varF2	Insight-sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	think, know, consider
F3	ratio of inferring & tentative tweets	Content	Text Sentiment Analysis	ratio	Measure each and aggregate by rumor/topic	YES	
varF3	Tentat-sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	may be, perhaps, guess

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F4	avg doubt LIWC	Content	Text Sentiment Analysis	avg	Measure each and aggregate by propagation tree	NO	$D = \frac{1}{\#\text{reposts}} \sum \frac{\#\text{doubt} - \#\text{non\_doubtt}}{\#\text{words in repost}}$
F5	avg surprise LIWC	Content	Text Sentiment Analysis	avg	Measure each and aggregate by propagation tree	NO	$S = \frac{1}{\#\text{reposts}} \sum \frac{\#\text{doubt} - \#\text{non\_doubtt}}{\#\text{words in repost}}$
F6	avg emoticon LIWC	Content	Text Sentiment Analysis	avg	Measure each and aggregate by propagation tree	NO	$E = \frac{1}{\#\text{reposts}} \sum \frac{\#\text{doubt} - \#\text{non\_doubtt}}{\#\text{words in repost}}$
varF6	contains emoticon smile	Content	Textual Patterns	boolean	Each tweet	NO	
varF6	contains emoticon frown	Content	Textual Patterns	boolean	Each tweet	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF6	fraction tweets emoticon smile	Content	Textual Patterns	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	
varF6	fraction tweets emoticon frown	Content	Textual Patterns	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	
F7	Posemo -sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	love, nice, sweet
varF7	sentiment positive words	Content	Text Sentiment Analysis	cnt	Measure each and aggregate by rumor/topic	NO	number of positive words in the text
varF7	# of positive words in microblogs	Content	Text Sentiment Analysis	cnt	Measure each and aggregate by timespan of event	YES	number of positive words in the text

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF7	% of positive microblogs	Content	Text Sentiment Analysis	fraction	Measure each and aggregate by timespan of event	YES	% positive words in the text
varF7	fraction sentiment positive	Content	Text Sentiment Analysis	fraction	Measure each and aggregate by rumor/topic	NO	The fraction of tweets with a positive score
varF7	presence of positive emotion words	Content	Text Sentiment Analysis	boolean	Each tweet	NO	true/false
F8	Social-sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	mate, talk, they, child
F9	Cogmech-sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	cause, know, ought

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F10	Excl- sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	but, without, exclude
F11	see - sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	view, saw, seen
F12	hear - sentiment LIWC	Content	Text Sentiment Analysis		Measure each and aggregate by rumor/topic	NO	listen, hearing
F13	All LIWC categories	Content	Text Sentiment Analysis	fraction	Measure each and aggregate by rumor/topic	YES	of words in message belongs to certain LIWC category
F14	Average sentiment score of microblogs	Content	Text Sentiment Analysis	avg	Measure each and aggregate by timespan of event	YES	$\frac{1}{\#\text{tweets}} \sum \text{pw} = \text{nw} + \text{pe} - \text{ne}$

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF14	sentiment score	Content	Text Sentiment Analysis	equation	Measure each and aggregate by rumor/topic	NO	Sum of 0.5 for weak positive/negative words, 1.0 for strong ones
varF14	average sentiment score	Content	Text Sentiment Analysis	avg	Measure each and aggregate by rumor/topic	NO	The average sentiment score of tweets
varF14	avg sentiment of reposts	Content	Text Sentiment Analysis	avg	Measure each and aggregate by propagation tree	NO	$= \frac{1}{\#\text{reposts}} \sum \frac{\#P - \#\text{Nt}}{\#\text{words in repost}}$
varF14	Sentiment For each	Content	Text Sentiment Analysis	equation	Measure each and aggregate by propagation tree	NO	
varF14	The six GPOMS sentiment dimensions (6 feats)	Content	Text Sentiment Analysis		Each meme (#,@,phrase,url)	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F15	average formality & sophistication of the tweets	Content	Text Sentiment Analysis	avg	Measure each and aggregate by rumor/topic	YES	online collection of vulgar words to collect a total of 349 vulgar words, collect a total of 362 emoticon, a binary feature indicating the presence of any of the 944 abbreviations, counting the number of characters in each word in the tweet and dividing by the total number of words in that tweet, depth of its dependency parse tree (Kong et al parser)
varF15	presence of swear words	Content	Text Sentiment Analysis	boolean	Each tweet	NO	
F16	sentiment labels - positive, neutral or negative for each tweet	Content	Text Sentiment Analysis	categorical	Measure each and aggregate by rumor/topic	NO	threeclass sentiment classification on a tweet. positive, neutral or negative
F17	patterns of enquiring or correcting behaviour	Content	Textual Patterns	boolean	Measure each and aggregate by signal pattern	NO	is (that | this | it) true wh[a]*t[!][1]* (real | really | unconfirmed) (rumor | debunk) (that | this | it) is not true

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF17	Percentage of signal tweets	Content	Textual Patterns	ratio	Measure each and aggregate by signal pattern	NO	
F18	length characters	Content	NLP	cnt	Measure each and aggregate by rumor/topic & Each tweet	NO	Length of the text of the tweet, in characters
varF18	average length	Content	NLP	avg	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	Average length of a tweet
varF18	average number of words per signal tweet	Content	NLP	avg	Each tweet	NO	average number of words per signal tweet
varF18	average number of words per any tweet in the cluster	Content	NLP	avg	Measure each and aggregate by signal pattern	NO	average number of words per tweet in the cluster

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF18	ratio of (average number of words per signal tweet) to (average number of words per any tweet in the cluster).	Content	NLP	ratio	Measure each and aggregate by signal pattern	NO	ratio of (average number of words per signal tweet) to (average number of words per any tweet in the cluster)
varF18	length words	Content	NLP	cnt	Measure each and aggregate by rumor/topic & Each tweet	NO	number of words in each tweet
F19	number of unique characters	Content	NLP	cnt	Each tweet	NO	number of unique characters
varF19	lexical diversity	Content	NLP	equation	Measure each and aggregate by rumor/topic	YES	For each rumor behavior category, we create a dictionary containing every unique word mentioned by tweets with that code

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F20	contains question mark	Content	Textual Patterns	boolean	Measure each and aggregate by rumor/topic	NO	Contains a question mark ’
varF20	fraction tweets question mark	Content	Textual Patterns	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	Contains a question mark ’
varF23	number of question marks	Content	Textual Patterns	cnt	Measure each and aggregate by rumor/topic	NO	Cnt ‘’
varF20	% of microblogs with multiple question marks	Content	Textual Patterns	fraction	Measure each and aggregate by timespan of event	YES	
varF20	contains multi quest	Content	Textual Patterns	boolean	Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F21	contains exclamation mark	Content	Textual Patterns	boolean	Measure each and aggregate by rumor/topic	NO	
varF21	contains multi excl	Content	Textual Patterns	boolean	Measure each and aggregate by rumor/topic	NO	
varF21	fraction tweets exclamation mark	Content	Textual Patterns	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	
varF21	number of exclamation marks	Content	Textual Patterns	cnt	Measure each and aggregate by rumor/topic	NO	
varF21	% of microblogs with multiple exclamation marks	Content	Textual Patterns	fraction	Measure each and aggregate by timespan of event	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F22	number of '!'	Content	Textual Patterns	cnt	Measure each and aggregate by rumor/topic	NO	
F23	contains stock symbol	Content	Textual Patterns	boolean	Measure each and aggregate by rumor/topic	NO	
varF23	fraction tweets stock symbol	Content	Textual Patterns	fraction	Measure each and aggregate by rumor/topic	NO	
varF23	presence of stock symbol	Content	Textual Patterns	boolean	Each tweet	NO	
F24	tweet contains 'via'	Content	Textual Patterns	boolean	Each tweet	NO	
varF24	client	Topic Propagation	Tweet Meta client	categorical	Measure each and aggregate by propagation tree	NO	the type of software client used to post the original message

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F25	presence of colon symbol	Content	Textual Patterns	boolean	Each tweet	NO	
F26	count uppercase letters	Content	NLP	cnt	Measure each and aggregate by rumor/topic	NO	
varF26	fraction tweets more than 30% uppercase	Content	NLP	fraction	Measure each and aggregate by rumor/topic	NO	
F27	contains pronoun first	Content	Grammar/ Syntax	boolean	Measure each and aggregate by rumor/topic & Each tweet	NO	
varF27	contains pronoun second	Content	Grammar/ Syntax	boolean	Measure each and aggregate by rumor/topic & Each tweet	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF27	contains pronoun third	Content	Grammar/ Syntax	boolean	Measure each and aggregate by rumor/topic & Each tweet	NO	
varF27	% of microblogs with the first-person pronouns	Content	Grammar/ Syntax	fraction	Measure each and aggregate by timespan of event	YES and NO	
F28	n-gram features	Content	Grammar/ Syntax		Measure each and aggregate by rumor/topic	YES and NO	
F29	part-of-speech patterns (unigram+bigram)	Content	Grammar/ Syntax		Measure each and aggregate by rumor/topic	YES and NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F30	topic type	Content	Topic Analysis	vector	Measure each and aggregate by propagation tree	NO	(LDA) model which returns an 18-topic
varF30	lda-based topic distribution of microblogs with 18 topics	Content	Topic Analysis	vector	Measure each and aggregate by timespan of event	YES	(LDA) model which returns an 18-topic
F31	entropy ratio	Content	Topic Analysis	ratio	Measure each and aggregate by signal pattern	NO	the ratio of the entropy of the word frequency distribution in the set of signal tweets to that in the set of all tweets in the cluster.
F32	number of urls	Topic Propagation	Tweet Meta URL	cnt	Measure each and aggregate by rumor/topic & Each tweet	NO	number of urls contained on a tweet

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF32	has url	Topic Propagation	Tweet Meta URL	boolean	Measure each and aggregate by propagation tree	NO	Whether the message contains URLs
varF32	fraction of tweets containing outside links	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	YES	%tweets with url= #tweets with url/#all tweets
varF32	fraction tweets url	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	NO	the fraction of tweets containing a url
varF32	% of microblogs with url	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by timespan of event	YES	
varF32	average number of URLs per signal tweet	Topic Propagation	Tweet Meta URL	avg	Measure each and aggregate by signal pattern	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF32	average number of URLs per any tweet in the cluster	Topic Propagation	Tweet Meta URL	avg	Measure each and aggregate by signal pattern	NO	
F33	contains popular domain top 100	Topic Propagation	Tweet Meta URL	boolean	Measure each and aggregate by rumor/topic	NO	contains a url whose domain is in 100 most popular
varF33	contains popular domain top 1000	Topic Propagation	Tweet Meta URL	boolean	Measure each and aggregate by rumor/topic	NO	one of the 1,000 most popular ones
varF33	contains popular domain top 10000	Topic Propagation	Tweet Meta URL	boolean	Measure each and aggregate by rumor/topic	NO	one of the 10,000 most popular ones
varF33	fraction popular domain top 100	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	NO	the fraction of tweets with a url in one of the top-100 domains

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF33	fraction popular domain top 1000	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	NO	in one of the top-1,000 domains
varF33	fraction popular domain top 10000	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	NO	in one of the top-10,000 domains
varF33	wot score for the url	Topic Propagation	Tweet Meta URL	score	Each tweet	NO	API https://www.mywot.com/
F34	count distinct expanded urls	Topic Propagation	Tweet Meta URL	cnt	Measure each and aggregate by rumor/topic	NO	the number of distinct urls found after expanding short urls
varF34	count distinct seemingly shortened urls	Topic Propagation	Tweet Meta URL	cnt	Measure each and aggregate by rumor/topic	NO	the number of distinct short urls
F35	share most frequent expanded url	Topic Propagation	Tweet Meta URL	fraction	Measure each and aggregate by rumor/topic	NO	the fraction of occurrences of the most frequent expanded url

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F36	Domain diversity	Topic Propagation	Tweet Meta URL	score	Measure each and aggregate by rumor/topic	YES	domains over time
F37	Top Domain volume	Topic Propagation	Tweet Meta URL	Cnt	Measure each and aggregate by rumor/topic	YES	number of top 10 domains over time
Fextra	html domfp	Topic Propagation	Tweet Meta URL			NO	we strip away all the content but the html 4.0 elements and then build a string by mapping html elements to symbols while preserving their order of appearance in the page.
varFextra	plagiarism	Topic Propagation	Tweet Meta URL			NO	google search with words count results
F38	url unigram	Topic Propagation	Tweet Meta URL	likelihood		NO	Given a set of training tweets, we fetch all the URLs in these tweets and build + and for unigrams (content of the URLs)
varF38	url bigram	Topic Propagation	Tweet Meta URL	likelihood		NO	Given a set of training tweets, we fetch all the URLs in these tweets and build + and for bigrams (content of the URLs)

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F39	share most frequent hashtag	Topic Propagation	Tweet Meta Hash	fraction		NO	the fraction of occurrences of the most frequent hashtag
F40	contains hashtag	Topic Propagation	Tweet Meta Hash	boolean	Measure each and aggregate by rumor/topic	NO	includes a hashtag
varF40	fraction tweets hashtag	Topic Propagation	Tweet Meta Hash	fraction	Measure each and aggregate by rumor/topic	YES and NO	The fraction of tweets containing hashtags
varF40	number hashtags	Topic Propagation	Tweet Meta Hash	cnt	Each tweet	NO	number of hashtags
varF40	average number of hashtags per signal tweet	Topic Propagation	Tweet Meta Hash	avg	Measure each and aggregate by signal pattern	NO	
varF40	number of hashtags per any tweet in the cluster	Topic Propagation	Tweet Meta Hash	avg	Measure each and aggregate by signal pattern	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F41	number distinct hashtags	Topic Propagation	Tweet Meta Hash	cnt	Measure each and aggregate by rumor/topic	NO	number of distinct hashtags
F42	log-likelihood ratio that for a given tweet, t, with a set of m hashtags	Topic Propagation	Tweet Meta Hash	likelihood	Each tweet	NO	we build two statistical models (+,), each showing the usage probability distribution of various hashtags.
F43	contains user mention	Topic Propagation	Tweet Meta @	Boolean	Measure each and aggregate by rumor/topic	NO	mentions a user: e.g. @cnnbrk
varF43	fraction tweets user mention	Topic Propagation	Tweet Meta @	fraction	Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF43	% of microblogs with @ mentions	Topic Propagation	Tweet Meta @	fraction	Measure each and aggregate by timespan of event	YES	
varF43	average number of usernames mentioned per signal tweet	Topic Propagation	Tweet Meta @	avg	Each tweet	NO	
varF43	Average number of usernames mentioned per any tweet in the cluster	Topic Propagation	Tweet Meta @	avg	Measure each and aggregate by signal pattern	NO	
varF43	count distinct users mentioned	Topic Propagation	Tweet Meta @	cnt	Measure each and aggregate by rumor/topic & Each tweet	NO	the number of distinct users mentioned in the tweets

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF43	share most frequent user mentioned	Topic Propagation	Tweet Meta @	Fraction	Measure each and aggregate by rumor/topic	NO	the fraction of user mentions of the most frequently mentioned user
F44	tweet is a reply	Topic Propagation	Tweet Meta @	boolean	Each tweet	NO	Twitter API
F45	controversiality	User meta	Influence	equation	Measure each and aggregate by rumor/topic	YES	controversiality = (p + n)^min(p/n,n/p)
F46	ratio of likes / dislikes for a youtube video	Topic Propagation	Tweet Meta URL	ratio	Each tweet	NO	Youtube API
F47	has multimedia	Topic Propagation	Tweet Meta multimedia	Boolean	Measure each and aggregate by propagation tree	NO	video, images
F48	source of tweet	Topic Propagation	Tweet Meta client	Categorical	Each tweet	NO	API

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F49	tweet contains geo-coordinates	Topic Propagation	Tweet Meta geolocation	Boolean	Each tweet	NO	API
F50	statuses count	User Meta	Role	cnt	Measure each and aggregate by rumor/topic	NO	the number of tweets at posting time
varF50	author average statuses count	User Meta	Role	avg	Measure each and aggregate by rumor/topic	NO	the average of author statuses count
varF50	average # of posts of users	User Meta	Role	avg	Measure each and aggregate by timespan of event	YES	
varF50	Average of number of tweets	User Meta	Role	avg	Measure each and aggregate by rumor/topic	YES	user influence

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF50	25th percentile of number of tweets	User Meta	Role	25th percentile	Measure each and aggregate by rumor/topic	YES	
varF50	median of number of tweets	User Meta	Role	median	Measure each and aggregate by rumor/topic	YES	
varF50	75th percentile of number of tweets	User Meta	Role	75th percentile	Measure each and aggregate by rumor/topic	YES	
varF50	Maximum of number of tweets	User Meta	Role	Maximum	Measure each and aggregate by rumor/topic	YES	
varF50	Standard deviation of number of tweets	User Meta	Role	stdev	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF50	Kurtosis deviation of number of tweets	User Meta	Role	kurtosis	Measure each and aggregate by rumor/topic	YES	
varF50	Skewness of number of tweets	User Meta	Role	skewness	Measure each and aggregate by rumor/topic	YES	
F51	originality	User Meta	Role	equation	Measure each and aggregate by rumor/topic	YES	originality =#tweets/#retweets
F52	engagement	User Meta	Influence	equation	Measure each and aggregate by rumor/topic	YES	engagement =(#tweets + #retweets + #replies + #favourites)/ account_age
F53	credibility	User Meta	Profile	cnt	Measure each and aggregate by rumor/topic/propagation tree/timespan of event	YES and NO	1 if verified, 0 otherwise

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF53	author fraction is verified	User Meta	Profile	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	the fraction of tweets from verified authors
varF53	% of verified users of each type	User Meta	Profile	fraction	Measure each and aggregate by rumor/topic/timespan of event	YES	e.g. celebrities
F54	registration age	User Meta	Profile	avg	Measure each and aggregate by rumor/topic/propagation tree & Each tweet	NO	the time passed since the author registered his/her account, in days
varF54	author average registration age	User Meta	Profile	avg	Measure each and aggregate by rumor/topic	NO	the average of author registration age

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF54	average days users' accounts exist since registration	User Meta	Profile	avg	Measure each and aggregate by rumor/topic/timespan of event	YES	
F55	has description	User Meta	Profile	Boolean	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	a non-empty 'bio' at posting time
varF55	% of users that provide personal description	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	
F56	% of users that provide personal picture in profile	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F57	Has url	User Meta	Profile	Boolean	Measure each and aggregate by rumor/topic	NO	
varF57	author fraction has url	User Meta	Profile	fraction	Measure each and aggregate by rumor/topic	NO	from authors with a non empty homepage url at posting time
F58	% of male users	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	
varF58	% of female users	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	
F59	% of users located in large cities	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	location where user was registered

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF59	% of users located in small cities	User Meta	Profile	fraction	Measure each and aggregate by timespan of event	YES	location where user was registered
varF59	Has location	User Meta	Profile	Boolean	Each tweet	NO	
F60	count followers	User Meta	Influence	cnt	Measure each and aggregate by rumor/topic/propagation tree & Each tweet	YES and NO	number of people following this author at posting time
varF60	author average count followers	User Meta	Influence	avg	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	of author count followers
varF60	Average of number of followers	User Meta	Influence	avg	Measure each and aggregate by rumor/topic	YES	user influence

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF60	25th percentile of number of followers	User Meta	Influence	25th percentile	Measure each and aggregate by rumor/topic	YES	
varF60	median of number of followers	User Meta	Influence	median	Measure each and aggregate by rumor/topic	YES	
varF60	75th percentile of number of followers	User Meta	Influence	75th percentile	Measure each and aggregate by rumor/topic	YES	
varF60	Maximum of number of followers	User Meta	Influence	Maximum	Measure each and aggregate by rumor/topic	YES	
varF60	Standard deviation of number of followers	User Meta	Influence	stdev	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF60	Kurtosis deviation of number of followers	User Meta	Influence	kurtosis	Measure each and aggregate by rumor/topic	YES	
varF60	Skewness of number of followers	User Meta	Influence	skewness	Measure each and aggregate by rumor/topic	YES	
F61	count friends	User Meta	Role	cnt	Measure each and aggregate by rumor/topic/propagation tree & Each tweet	YES and NO	number of people this author is following at posting time
varF61	author average count friends	User Meta	Role	avg	Measure each and aggregate by rumor/topic/timespan of event	YES and NO	of author count friends

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF61	average reputation score of users	User Meta	Role	avg	Measure each and aggregate by timespan of event	YES	followers/followees ratio
varF61	role	User Meta	Role	equation	Measure each and aggregate by rumor/topic	YES	role = #followers/#followees
varF61	Average of number of friends	User Meta	Role	avg	Measure each and aggregate by rumor/topic	YES	
varF61	25th percentile of number of friends	User Meta	Role	25th percentile	Measure each and aggregate by rumor/topic	YES	
varF61	median of number of friends	User Meta	Role	median	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF61	75th percentile of number of friends	User Meta	Role	75th percentile	Measure each and aggregate by rumor/topic	YES	
varF61	Maximum of number of friends	User Meta	Role	Maximum	Measure each and aggregate by rumor/topic	YES	
varF61	Standard deviation of number of friends	User Meta	Role	stdev	Measure each and aggregate by rumor/topic	YES	
varF61	Kurtosis deviation of number of friends	User Meta	Role	kurtosis	Measure each and aggregate by rumor/topic	YES	
varF61	Skewness of number of friends	User Meta	Role	skewness	Measure each and aggregate by rumor/topic	YES	
F62	Ratio of statuses to followers	User Meta	Influence	ratio	Each tweet	NO	Statuses/followers

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F63	count tweets	Topic Propagation	Tweet Volume	cnt	Measure each and aggregate by rumor/topic	NO	number of tweets
varF63	# of microblogs	Topic Propagation	Tweet Volume	cnt	Measure each and aggregate by timespan of event	YES	
F64	day weekday	Topic Propagation	Tweet Meta timestamp	timespan	Measure each and aggregate by rumor/topic	NO	the day of the week in which this tweet was written
F65	count distinct authors	Topic Propagation	User Volume	cnt	Measure each and aggregate by rumor/topic	NO	the number of distinct authors of tweets
varF65	share most frequent author	Topic Propagation	User Volume	fraction	Measure each and aggregate by rumor/topic	NO	the fraction of tweets authored by the most frequent author

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F66	number of retweets	Topic Propagation	Tweet Meta Reposts	cnt	Each tweet	NO	
varF66	Is a retweet	Topic Propagation	Tweet Meta Reposts	Boolean	Measure each and aggregate by rumor/topic & Each tweet	NO	is a retweet: contains 'rt '
varF66	num of reposts	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by timespan of event	YES	
varF66	fraction retweets	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by rumor/topic	NO	the fraction of tweets that are re-tweets
varF66	average # of retweets	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by timespan of event	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF66	average # of comments for weibo posts	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by timespan of event	YES	
varF66	percentage of retweets among the signal tweets	Topic Propagation	Tweet Volume	avg	Each tweet	NO	
varF66	percentage of retweets among all tweets in the cluster	Topic Propagation	Tweet Volume	avg	Measure each and aggregate by signal pattern	NO	
F67	repost time	Topic Propagation	Tweet Meta timestamp	equation	Measure each and aggregate by propagation tree	NO	
F68	num of comments	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by timespan of event	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F69	search engine	Tweet Text	search	score	Measure each and aggregate by propagation tree	NO	number of results returned original message and "false rumor"
F70	percentage of signal tweets (1 feature)	Topic Propagation	Tweet Volume	fraction	Measure each and aggregate by signal pattern	NO	the ratio of signal tweets to all tweets in the cluster.
F71	number of seconds since the tweet	Topic Propagation	Tweet Meta timestamp	timespan	Each tweet	NO	
F72	fraction of low-to-high diffusion	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES and NO	from a sender with lower influence to a receiver with higher influence %low-high diffusion =#low-high diffusions/#all diffusion events
varF72	fraction of htl among information diffusion	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F73	fraction of nodes in largest connected component (lcc) of diffusion network	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	%nodes in lcc_f= #nodes in lcc/#all nodes
F74	fraction of nodes in largest connected component (lcc) of the friendship network	Network Structure	Community	fraction	Measure each and aggregate by rumor/topic	NO	
varF74	# nodes in lcc of f	Network Structure	Community	cnt	Measure each and aggregate by rumor/topic	YES and NO	
varF74	# edges in lcc of f	Network Structure	Community	cnt	Measure each and aggregate by rumor/topic	YES and NO	

Feature cluster	Feature name	Type	Type of Measurement analysis	Granularity	Time	Definition
varF74	average degree of nodes in lcc of f	Network Structure	Communityavg	Measure each and aggregate by rumor/topic	YES	
varF74	average clustering coefficients of lcc of f	Network Structure	Communityavg	Measure each and aggregate by rumor/topic	YES and NO	
varF74	density of lcc of f	Network Structure	Communitygraph	Measure each and aggregate by rumor/topic	YES and NO	
varF74	median in-degree in the lcc	Network Structure	Communitygraph	Measure each and aggregate by rumor/topic	NO	
varF74	median out-degree in the lcc	Network Structure	Communitygraph	Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF74	number of nodes in the friendship network	Network Structure	Potential Impact	cnt	Measure each and aggregate by rumor/topic	YES and NO	
varF74	number of links in the friendship network	Network Structure	Potential Impact	cnt	Measure each and aggregate by rumor/topic	YES and NO	
varF74	density of the friendship network	Network Structure	Potential Impact	graph	Measure each and aggregate by rumor/topic	NO	
varF74	clustering coefficient of the friendship network	Network Structure	Community	graph	Measure each and aggregate by rumor/topic	NO	
varF74	median in-degree of the friendship network	Network Structure	Friendship Reciprocity	graph	Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF74	median out-degree of the friendship network	Network Structure	Friendship Reciprocity	graph	Measure each and aggregate by rumor/topic	NO	
varF74	# nodes without incoming fdges in f	Network Structure	Friendship Reciprocity	cnt	Measure each and aggregate by rumor/topic	YES	
varF74	# nodes without outgoing fdges in f	Network Structure	Friendship Reciprocity	cnt	Measure each and aggregate by rumor/topic	YES	
varF74	# isolated nodes in f	Network Structure	Friendship Reciprocity	cnt	Measure each and aggregate by rumor/topic	YES	
varF74	percent of nodes without incoming edges in f	Network Structure	Friendship Reciprocity	fraction	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF74	percent of nodes without outgoing fdges in f	Network Structure	Friendship Reciprocity	fraction	Measure each and aggregate by rumor/topic	YES	
varF74	percent of isolated nodes in f	Network Structure	Friendship Reciprocity	fraction	Measure each and aggregate by rumor/topic	YES	
F75	# nodes in d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# edges in d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# nodes without incoming ddges in d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF75	# nodes without outgoing ddges in d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# isolated nodes in d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	percent of nodes without incoming ddges in d	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	
varF75	percent of nodes without outgoing ddges in d	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	
varF75	percent of isolated nodes in d	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES and NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF75	# nodes in lcc of d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# edges in lcc of d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	average degree of nodes in lcc of d	Topic Propagation	Flow	avg	Measure each and aggregate by rumor/topic	YES	
varF75	average clustering coefficients of lcc of d	Topic Propagation	Flow	avg	Measure each and aggregate by rumor/topic	YES	
varF75	density of lcc of d	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF75	# nodes in e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	extended directed subgraph of users, who have either posted any tweet related to the event or who is following or is followed by other participating users
varF75	# edges in e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# nodes without incoming edges in e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# nodes without outgoing edges in e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# isolated nodes in e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF75	percent of nodes without incoming edges in e	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	
varF75	percent of nodes without outgoing edges in e	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	
varF75	percent of isolated nodes in e	Topic Propagation	Flow	fraction	Measure each and aggregate by rumor/topic	YES	
varF75	# nodes in lcc of e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	
varF75	# edges in lcc of e	Topic Propagation	Flow	cnt	Measure each and aggregate by rumor/topic	YES	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF75	average degree of nodes in lcc of e	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	YES	
varF75	average clustering coefficients of lcc of e	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	YES	
varF75	density of lcc of e	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	YES	
F76	propagation initial tweets	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	the degree of the root in a propagation tree
varF76	propagation max sub-tree	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	the total number of tweets in the largest sub-tree of the root, plus one

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF76	propagation max degree	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	the maximum degree of a node that is not the root
varF76	propagation avg degree	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	the average degree of a node that is not the root (2 feat.)
varF76	propagation max depth	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	the depth of a propagation tree (0=empty tree, 1=only initial tweets2=only retweets of the root)
varF76	propagation avg depth	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	per-node average depth of propagation tree
varF76	propagation max level	Topic Propagation	Flow	graph	Measure each and aggregate by rumor/topic	NO	The max. size of a level in the propagation tree (except children of root)

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
F77	average depth to breadth ratio	Topic Propagation	Flow	avg	Measure each and aggregate by rumor/topic	YES	$A = \frac{\sum \frac{\#\text{ nodes in largest chain}}{\text{All nodes}}}{N}$
F78	ratio of new users	Topic Propagation	Flow	ratio	Measure each and aggregate by rumor/topic	YES	%new users(ti) = new users(ti)/users(ti)
F79	Ratio of original tweets	Topic Propagation	Flow	ratio	Measure each and aggregate by rumor/topic	YES	%Original Tweets = #Tweets + #Replies / #Tweets + #Replies + #Retweets
F79	log-likelihood ratio that ui is under a positive user model			likelihood		NO	Given a set of training instances, we build a positive and a negative user models. The first model is a probability distribution over all users that have posted a positive instance or have been retweeted in a positive instance.

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF79	log-likelihood ratio that the tweet is re-tweeted from a user (uj) who is under a positive user model than a negative user model			likelihood		NO	Intuitively, t is more likely to be a rumor if (1) uj has a history of posting or re-tweeting rumors, or (2) ui has posted or retweeted rumors in the past. The distinction between the posting user and the re-tweeted user is important, since some times the users modify the re-tweeted message in a way that changes its meaning and intent
F80	Strength of external shock at birth	Topic Propagation	Tweet Volume	score	Measure each and aggregate by rumor/topic	NO	
varF80	Periodicity of external shock	Topic Propagation	Tweet Volume	score	Measure each and aggregate by rumor/topic	NO	
varF80	External shock periodicity offset	Topic Propagation	Tweet Volume	score	Measure each and aggregate by rumor/topic	NO	

Feature cluster	Feature name	Type	Type of analysis	Measurement	Granularity	Time	Definition
varF80	Interaction periodicity offset	Topic Propagation	Tweet Volume	score	Measure each and aggregate by rumor/topic	NO	
varF80	strength of interaction periodicity	Topic Propagation	Tweet Volume			NO	

Danae Pla Karidi

Contact information:

- Information Management Systems Institute (IMSI)
  Athena Research Center
  Artemidos 6 Epidavrou
  Marousi 15125, Greece
  tel: (+30) 2106875403
  e-mail: danae@athenarc.gr

- Knowledge and Database Systems Laboratory (KDBSL)
  School of Electrical and Computer Engineering
  National Technical University of Athens
  Iroon Polytechniou 9, Zografou Campus
  Zografou 15780, Greece
  tel: (+30) 2107721402

Academic Qualifications:

- 2014–2021: PhD student, School of Electrical and Computer Engineering,
  National Technical University of Athens,
  supervisor: Prof. Yannis Vassiliou

- 2007–2014: Diploma of Electrical and Computer Engineer,
  School of Electrical and Computer Engineering
  National Technical University of Athens
  (Graduation Grade “Very Good”)

Affiliations and memberships:

- Member of the Technical Chamber of Greece
- Member of the Greek Electrical and Mechanical Engineer Association

Academic interests:

- Social network analysis
- Knowledge graph and semantic network mining
- Information diffusion and mining in social networks

Work experience:

- 2014-2021: Research associate, Information Management Systems Institute, Athena Research Center
- 2014: Research assistant, Institute of Communication and Computer Systems, National Technical University of Athens

Research publications

Peer-reviewed journals:

- Pla Karidi, D., Stavrakas, Y., Vassiliou, Y. (2018). Tweet and followee personalized recommendations based on knowledge graphs. Journal of Ambient Intelligence and Humanized Computing, 9, 2035-2049.

- Pla Karidi, D., Nakos, H., Efentakis, A., Stavrakas, Y. (2017). CitySense: Combining Geolocated Data for Urban Area Profiling. International Journal on Advances in Software (2017): v 10 n 34.

- Vergoulis, T., Kanellos, I., Chatzopoulos, S., Karidi, D.P., Dalamagas, T. (2021). BIP4COVID19: Releasing impact measures for articles relevant to COVID-19. Special Issue of Quantitative Science Studies (QSS) on "Scientific Knowledge Graphs and Research Impact Assessment" .

- Pla Karidi, D., Nakos, H., Stavrakas, Y., Vassiliou, Y. (2021). PHONY: Automatic Generation of Feature-Agnostic Datasets for Fake News Detection in Social Media. Submitted to SN Computer Science journal.

Peer-reviewed conference proceedings:

- Pla Karidi, D. (2016). From user graph to Topics Graph: Towards twitter followee recommendation based on knowledge graphs. 2016 IEEE 32nd International Conference on Data Engineering Workshops (ICDEW), 121-123.

- Pla Karidi, D., Stavrakas, Y., Vassiliou, Y. (2016). A Personalized Tweet Recommendation Approach Based on Concept Graphs. 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 253-260.

- Pla Karidi, D., Nakos, H., Efentakis, A., Stavrakas, Y. (2017). CitySense: Retrieving, Visualizing and Combining Datasets on Urban Areas. DBKDA 2017, 70.

- Pla Karidi, D., Nakos, H., Stavrakas, Y. (2019). Automatic Ground Truth Dataset Creation for Fake News Detection in Social Media. IDEAL.

- Pla Karidi, D., Koutrika, G., Stavrakas, Y. (2021) KNOwDE: Knowledge-based Recommendations for Data Exploration. Under submission.

Appendix B

Extended Greek Abstract

Εκτεταμένη περίληψη στα Ελληνικά

Αξιοποίηση κοινωνικών δικτύων και γράφων γνώσης για την εύρεση και σύσταση ενδιαφέρουσας και αξιόπιστης πληροφορίας

Β′.1 Εισαγωγή

Το κύριο ενδιαφέρον της έρευνάς μας ήταν η δημιουργία ενδιαφερουσών συστάσεων και την ανακάλυψη αξιόπιστης πληροφορίας που διαδίδεται στα κοινωνικά δίκτυα. Για το σκοπό αυτό, εστιάσαμε στην κατανόηση του ρόλου του σημασιολογικού περιεχομένου, των ιδιοτήτων των γράφων και των προτύπων διάχυσης της πληροφορίας στα κοινωνικά δίκτυα.

Πώς μπορούμε να προτείνουμε ενδιαφέρον περιεχόμενο στους χρήστες κοινωνικών δικτύων; Πώς μπορούμε να εντοπίσουμε έγκαιρα τη διάδοση των ψευδών ειδήσεων σε ένα δίκτυο που εξελίσσεται στο χρόνο; Πώς μπορούμε να ξεπεράσουμε την ασάφεια της φυσικής γλώσσας για να εξερευνήσουμε αποτελεσματικά μεγάλα σύνολα δεδομένων; Πώς μπορούμε να αξιοποιήσουμε τα κοινωνικά δίκτυα για να ανακαλύψουμε αξιόπιστες πληροφορίες;

Οι απαντήσεις σε τέτοια ερωτήματα είναι ζωτικής σημασίας για μια σειρά από τομείς εφαρμογής, από την ανάπτυξη νέων αλγορίθμων για τη σύσταση ελκυστικού περιεχομένου σε κοινωνικά δίκτυα και άλλες πλατφόρμες, την παρακολούθηση της διάδοσης ψευδών ειδήσεων, την ανάπτυξη τεχνικών εξερεύνησης δεδομένων που βασίζονται στη γνώση, μέχρι την ανακάλυψη και την κατάταξη αναδυόμενων άρθρων βιβλιογραφίας με βάση την προσοχή που λαμβάνουν στα μέσα κοινωνικής δικτύωσης.

Μια βασική διαίσθηση πίσω από τη μελέτη μας ήταν η χρήση γράφων γνώσης ως το στέρεο γνωστικό υπόβαθρο για την εκμετάλλευση των σημασιολογικών σχέσεων μεταξύ των ενδιαφερόντων των χρηστών και των αντικειμένων δεδομένων. Στο δρόμο προς την υλοποίηση του σημασιολογικού ιστού, οι μηχανικοί γνώσης οργανώνουν την ανθρώπινη γνώση με τυπικό και, ταυτόχρονα, ουσιαστικό τρόπο. Σε αυτό το πλαίσιο, οι γράφοι γνώσης είναι μια σημαντική δομή για την οργάνωση των αντικειμένων της ανθρώπινης γνώσης και των σχέσεων μεταξύ τους, με σημασιολογικά δομημένες πληροφορίες, έτσι ώστε τα υπολογιστικά συστήματα να μπορούν να τη διαχειρίζονται. Δεν είναι τυχαίο ότι οι εταιρείες με τη μεγαλύτερη επιρροή στον τομέα της πληροφορικής προωθούν την ανάπτυξη τέτοιων δομών για να βελτιώσουν τις εμπορικές τους υπηρεσίες (πλοήγηση, αναζήτηση, εξατομικευμένες

προσφορές, στοχευμένη διαφήμιση) και να αυξήσουν την αφοσίωση των χρηστών. Οι κόμβοι των γράφων γνώσης αντιπροσωπεύουν συγκεκριμένα αντικείμενα γνώσης και οι ακμές αντιπροσωπεύουν τις σχέσεις μεταξύ τους. Δηλαδή, επικεντρώνονται στην αναπαράσταση συγκεκριμένων γεγονότων, ανθρώπων, οντοτήτων, αντικειμένων, περιοχών και σχέσεων μεταξύ τους. Σε αυτή τη διατριβή, διερευνήσαμε τις δυνατότητες σύνδεσης μεταξύ αυτών των ερευνητικών περιοχών.

- **Συστάσεις ενδιαφέροντος περιεχομένου στα κοινωνικά δίκτυα**

  Η αυξανόμενη χρήση των μέσων κοινωνικής δικτύωσης έχει επιφέρει αλλαγές στον τρόπο με τον οποίο ένα μεγάλο μέρος της ανθρωπότητας επικοινωνεί, ενημερώνεται, εργάζεται, διαμορφώνει την αντίληψή του για τον κόσμο και τα κοινωνικά φαινόμενα. Για παράδειγμα, οι χρήστες του Twitter δημιουργούν εκατοντάδες εκατομμύρια tweets κάθε μέρα, δημιουργώντας έναν τεράστιο όγκο διαθέσιμων πληροφοριών που καταλήγουν στα χρονοδιαγράμματα άλλων χρηστών. Εξαιτίας αυτής της υπερφόρτωσης περιεχομένου, οι χρήστες συχνά κουράζονται να περιηγούνται ψάχνοντας για ενδιαφέροντα tweets και ως εκ τούτου χάνουν ενδιαφέροντα tweets. Μια λύση σε αυτό το πρόβλημα μπορεί να είναι η ανάπτυξη αποτελεσματικών συστημάτων συστάσεων που βοηθούν τους χρήστες να φιλτράρουν τα μη ενδιαφέροντα τωεετς και να προτείνουν ταξινομημένα tweets και χρήστες με σχετικά ενδιαφέροντα, με αποτέλεσμα ένα πιο ελκυστικό χρονοδιάγραμμα.

  Όταν αναλύουν τη διάδοση του περιεχομένου των κοινωνικών δικτύων, οι επιστήμονες αντιμετωπίζουν ένα σοβαρό πρόβλημα: ο αλγόριθμος που ταξινομεί τις δημοσιεύσεις στο χρονοδιάγραμμα των χρηστών είναι συνήθως άγνωστος. Ωστόσο, ο ρόλος του στη διαδικασία διάχυσης πληροφοριών είναι κρίσιμος. Επιπλέον, η ροή πληροφοριών σχετικά με τη δραστηριότητα των χρηστών είναι ένα εμπόρευμα που χρησιμοποιείται για την αποτελεσματική έκθεση μιας ομάδας-στόχου σε ένα συγκεκριμένο εμπορικό μήνυμα. Έτσι, οι πλατφόρμες μέσων κοινωνικής δικτύωσης μοντελοποιούν τη συμπεριφορά των χρηστών, προβλέπουν και μαθαίνουν από αυτήν, εκμεταλλεύονται τις προτιμήσεις, τα ενδιαφέροντα και τις συνήθειες των ανθρώπων, τους προσφέρουν ενδιαφέρον περιεχόμενο και έτσι αυξάνουν την αφοσίωση στην πλατφόρμα και το εμπορικό περιεχόμενο.

  Επιπλέον, οι αλγόριθμοι συστάσεων των υπηρεσιών κοινωνικής δικτύωσης δεν είναι στατικοί. Αντίθετα, γίνονται όλο και πιο περίπλοκοι με βάση την έρευνα και την ανάπτυξη νέων μοντέλων συστάσεων και τεχνικών δημιουργίας προφίλ χρήστη. Ωστόσο, η κλειστότητά τους καθιστά ασαφές εάν χρησιμεύουν για να παρέχουν ¨αυτό που ενδιαφέρεται να δει κάθε χρήστης' ή αν είναι περισσότερο προσανατολισμένοι στο να δώσουν προτεραιότητα στο ¨τι πρέπει να δει κάθε χρήστης' για να ανταποκριθούν σε διάφορες απαιτήσεις. Επιπλέον, είναι αδύνατο να αξιολογηθεί η διαδικασία εξατομίκευσης που ακολουθούν και το κατά πόσο αυτή διευκολύνει την επέκταση φαινομένων παραπλάνησης (fake news, bots, trolls κ.λπ.). Επομένως, η εξέλιξη των αλγορίθμων συστάσεων παρουσιάζει μεγάλο ενδιαφέρον και ένα τέτοιο χαρακτηριστικό παράδειγμα είναι ο αλγόριθμος του Twitter. Ο αλγόριθμος του Twitter, όπως και οι περισσότεροι αλγόριθμοι μέσων κοινωνικής δικτύωσης, βασίζεται στην εξατομίκευση και τη μηχανική μάθηση για την κατάταξη του περιεχομένου με βάση ορισμένα σήματα: χρόνος (πρόσφατο προς παλαιότερο), συνάφεια, αφοσίωση, εμπλουτισμένα μέσα και η εξελισσόμενη παρελθοντική συμπεριφορά του χρήστη (like, click, retweets, δημοτικότητα, περιοχή κ.λπ.). Παρόλο που τέτοιοι αλγόριθμοι κατευθύνουν και διοχετεύουν την προσοχή εκατομμυρίων χρηστών, εξακολουθούμε να γνωρίζουμε λίγα για αυτούς.

  Σε αυτό το πλαίσιο, οι αλγόριθμοι συστάσεων ενδέχεται να περιορίσουν την προοπτική του χρήστη προτείνοντας περιεχόμενο παρόμοιο με αυτό που συνήθως δημοσιεύει ή με

το οποίο ασχολείται. Ως εκ τούτου, οι χρήστες εκτίθενται σε συγκεκριμένες πληροφορίες και αλληλεπιδρούν με ειδήσεις που πιθανότατα προωθούν τις αγαπημένες τους απόψεις, με αποτέλεσμα τη δημιουργία κοινωνικών ομάδων με ομοϊδεάτες (echo chambers). Το φαινόμενο *echo chamber* επηρεάζει τον τρόπο με τον οποίο καταναλώνονται οι ειδήσεις και ενισχύει τη διάδοση παραπλανητικού περιεχομένου, λειτουργώντας ως κόμβοι ενίσχυσης ψευδών ειδήσεων.

- **Ανακάλυψη αξιόπιστων πληροφοριών στα κοινωνικά δίκτυα** Η ευρεία χρήση των μέσων κοινωνικής δικτύωσης ως πηγή ειδήσεων και πληροφοριών έχει θέσει ορισμένες νέες προκλήσεις. Αν και τα φαινόμενα παραπληροφόρησης υπάρχουν από τη γέννηση του έντυπου Τύπου, οι νέες διαδικτυακές πλατφόρμες επιταχύνουν και ενισχύουν τη διάδοσή τους, θέτοντας νέα προβλήματα και προκλήσεις. Ωστόσο, σε αντίθεση με τους παραδοσιακούς οργανισμούς ειδήσεων, οι πλατφόρμες μέσων κοινωνικής δικτύωσης επιτρέπουν σε εκατομμύρια χρήστες να παράγουν και να έχουν πρόσβαση σε μια τεράστια πηγή πληροφοριών ελεύθερα. Με βάση την πολύ ταχύτερη διάχυση των ειδήσεων, τη σχετική ανωνυμία που προσφέρουν τα μέσα κοινωνικής δικτύωσης και τους φανερούς και κρυφούς μηχανισμούς που λειτουργούν στο διαδίκτυο εξυπηρετώντας πολιτικά, οικονομικά, γεωπολιτικά και άλλα συμφέροντα, το φαινόμενο της διάδοσης ψευδών ειδήσεων έχει εξελιχθεί και αποκτήσει ιδιαίτερα χαρακτηριστικά που σχετίζονται με τη διάδοση στα μέσα κοινωνικής δικτύωσης.

  Η καταπολέμηση των ψευδών ειδήσεων είναι ένα θέμα στο οποίο πρέπει να επικεντρωθούν οι μηχανικοί των κοινωνικών δικτύων. Ωστόσο, οι πηγές και τα αίτια της παραπληροφόρησης εδράζονται στο φάσμα των κοινωνικών φαινομένων. Αυτό εξηγεί γιατί πολλές φήμες επιβιώνουν για μεγάλο χρονικό διάστημα μετά την αποκάλυψή τους, γιατί παρά την άνοδο του μορφωτικού επιπέδου της ανθρωπότητας, επιβιώνουν αντιεπιστημονικές θεωρίες και προκαταλήψεις. Ο λόγος είναι ότι ο άνθρωπος μαθαίνει σε ένα κοινωνικό πλαίσιο, υιοθετεί και κατασκευάζει ένα μεγάλο μέρος των αντιλήψεών του με βάση τα δεδομένα που του δίνουν ως αδιαμφισβήτητη γνώση άνθρωποι και πηγές που εμπιστεύεται (δάσκαλοι, γονείς, φίλοι, πηγές πληροφοριών κ.λπ.). Σε ορισμένες περιπτώσεις, η παραπληροφόρηση ξεκινά από συγκεκριμένες ειδησεογραφικές σελίδες και μέσω της προώθησης των μέσων κοινωνικής δικτύωσης, βρίσκει θέση ακόμη και στις ιστοσελίδες ειδησεογραφικών οργανισμών που θεωρούνται γενικώς αξιόπιστοι.

  Επιπλέον, οι κρατικές υπηρεσίες πληροφοριών και οι περισσότεροι στρατοί έχουν ειδικές διευθύνσεις και τμήματα που εμπλέκονται στον κυβερνοπόλεμο, κεντρική πτυχή του οποίου είναι η προπαγάνδα και η διάδοση παραπληροφόρησης για τη σύγχυση και την εξαπάτηση του εχθρού. Έτσι, μερικές φορές πίσω από τις ψευδείς ειδήσεις κρύβεται ένας «πόλεμος» κρατικών και επιχειρηματικών συμφερόντων, ένας «πόλεμος» που δεν εξυπηρετεί μόνο τους σκοπούς της προπαγάνδας αλλά έχει και σοβαρές οικονομικές και κοινωνικές συνέπειες. Σε κάθε περίπτωση, υπάρχουν πολλές πρόσφατες δημοσιεύσεις [172, 131, 25], σύμφωνα με τις οποίες δεδομένα κοινωνικής δικτύωσης συγκεντρώνονται στα χέρια κυβερνήσεων και ιδιωτικών εταιρειών, διαμορφώνοντας τις απαιτήσεις για την ύπαρξη συστημάτων μαζικής παρακολούθησης. Με βάση τα παραπάνω, είναι επόμενο τα μέσα κοινωνικής δικτύωσης να χρησιμοποιούνται ως πολιτικά εργαλεία για τη διαμόρφωση της λεγόμενης «κοινής γνώμης», χειραγωγώντας συνειδήσεις, ουσιαστικά, για να αυξήσουν την επιρροή κρατών, πολιτικών κομμάτων, επιχειρηματικών ομάδων. Στο πλαίσιο αυτό, οι αντίπαλοι «στρατοί» αποκαλούν συχνά τη δραστηριότητα του αντιπάλου «παραπληροφόρηση» ή «ρητορική μίσους», με αποτέλεσμα να προκύπτουν πολλές περιπτώσεις λογοκρισίας στο όνομα «καταπολέμησης της παραπληροφόρησης και της ρητορικής μίσους». Επιπλέον, πολλοί οργανισμοί και υπηρεσίες ελέγχου γεγονότων έχουν κατηγορηθεί για μεροληψία, ενώ το αποτέλεσμα του ελέγχου γεγονότων μπορεί να ποικίλλει εξαιτίας διάφορων παραγόντων [99].

Ωστόσο, η ανάπτυξη μεθόδων και αλγορίθμων που μπορούν να ανιχνεύσουν ύποπτο και παραπλανητικό περιεχόμενο και να παράγουν σήματα έγκαιρης ανίχνευσης αποτελεί σημαντική συμβολή στον περιορισμό και την καταπολέμηση της περαιτέρω διάδοσης τέτοιου περιεχομένου. Ωστόσο, κανένα σύστημα δεν μπορεί να θεραπεύσει μια επιδημία παραπληροφόρησης, καθώς οι πηγές και τα αίτια του φαινομένου της παραπληροφόρησης παραμένουν κοινωνικές και όχι τεχνικές.

Εκτός από τον τεράστιο όγκο δεδομένων που παράγονται λόγω της δραστηριότητας στα κοινωνικά δίκτα, τα μεγάλα δεδομένα οδηγούν σε σημαντική αύξηση των δεδομένων και, παρά τη σχετικά μικρότερη διαθεσιμότητά τους, αυξάνουν την ανάγκη για αποτελεσματικά συστήματα εξερεύνησής τους, καθώς αυτή μπορεί να αναδείξει νέες δυνατότητες για την ανάπτυξη της παραγωγής, την οργάνωση της εργασίας, την πρόοδο της επιστήμης. Το μέγεθος και η ποικιλομορφία των διαθέσιμων συνόλων δεδομένων, η συνεχής συλλογή και παραγωγή τους από συστήματα, αισθητήρες, επιστήμονες, οργανισμούς, κράτη και η ανάπτυξη του διαδικτυακού περιεχομένου έχουν ως αποτέλεσμα ένα πλούσιο και ετερογενές περιεχόμενο. Ως εκ τούτου, ο όγκος και η πολυπλοκότητα των διαθέσιμων δεδομένων καθιστούν τα αποτελεσματικά συστήματα εξερεύνησης δεδομένων όλο και πιο σημαντικά.

Σε αυτή τη διατριβή, μελετήσαμε τις συνδέσεις αυτών των ερευνητικών περιοχών, που βασίζονται στο συμπαγές σημασιολογικό υπόβαθρο των γράφων γνώσης. Για παράδειγμα, ο συνδυασμός της γνώσης που παρέχεται από τους πιο προηγμένους γράφους γνώσης και των δεδομένων από κοινωνικά δίκτυα μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα των συστημάτων συστάσεων. Επιπλέον, οι σχέσεις που εξάγονται από βάσεις γνώσεων, όπως οι υπηρεσίες ελέγχου γεγονότων, μπορούν να παρέχουν σύνολα δεδομένων εκπαίδευσης για την ανάπτυξη αποτελεσματικών μοντέλων ανίχνευσης παραπληροφόρησης και με τη σειρά τους, αυτά τα μοντέλα ανίχνευσης μπορούν να βοηθήσουν στον καθαρισμό των υπαρχόντων γράφων γνώσης από λάθη και ψευδείς πληροφορίες. Επιπλέον, τα σημασιολογικά στοιχεία των γράφων γνώσης μπορούν να βοηθήσουν στην αποτελεσματική εξερεύνηση των συνόλων δεδομένων και η ανάλυση δεδομένων από κοινωνικά δίκτυα μπορεί να συμβάλει στην περιήγηση και την κατανόηση των διαθέσιμων δεδομένων. Επιπλέον, στη διατριβή παρουσιάζονται και εφαρμογές που αναπτύχθηκαν στα πλαίσιά της και άπτονται αυτών των περιοχών.

Β′.2 Επισκόπηση διατριβής

Β′.2.1 Εξατομικευμένες προτάσεις για tweet και followee με βάση γράφους γνώσης

Η σύσταση περιεχομένου και συνδέσεων στα κοινωνικά δίκτυα αντιμετωπίζει πολλαπλές προκλήσεις. Πρώτον, η δημιουργία του προφίλ χρήστη συχνά υπονομεύεται από τους περιορισμούς που τίθενται από τις υπηρεσίες κοινωνικής δικτύωσης σχετικά με τη διαθεσιμότητα των δεδομένων δικτύου των χρηστών. Επομένως, οι τεχνικές που κατασκευάζουν προφίλ με βάση τις κοινωνικές συνδέσεις χρηστών (συνεργατικό φιλτράρισμα) απαιτούν να ανακτηθεί μεγάλος όγκος δεδομένων δικτύου, να αποθηκευθεί και να αναλυθεί, και επομένως δεν μπορούν να επικαιροποιούνται αποτελεσματικά με την εμφάνιση νέων tweets στη ροή. Επιπλέον, οι προσεγγίσεις συνεργατικού φιλτραρίσματος πιθανότατα θα αξιολογήσουν παρόμοια αντικείμενα διαφορετικά, εάν αυτά δημοσιεύονται από διαφορετικούς χρήστες, παρόλο που έχουν το ίδιο περιεχόμενο. Αυτές οι προσεγγίσεις απαιτούν επίσης κάθε στοιχείο να λαμβάνει άμεση ανατροφοδότηση από πολλούς χρήστες προτού να συσταθεί σε άλλους, γνωστό ως πρόβλημα ¨ψυχρής εκκίνησης¨. Μια διαφορετική προσέγγιση, οι συστάσεις που βασίζονται στο περιεχόμενο, βασίζεται στην ομοιότητα του κειμένου. Ωστόσο, στερείται αποτελεσματικότητας στην περίπτωση των υπηρεσιών microblogging λόγω του μικρού μεγέθους του κειμένου.

Η άποψή μας είναι ότι τα σημασιολογικά χαρακτηριστικά των γράφων γνώσης μπορούν να είναι χρήσιμα για τη δημιουργία προφίλ χρήστη και την ανάλυση δεδομένων κοινωνικών δικτύων. Ωστόσο, στην περίπτωση του Twitter, το περιεχόμενο των tweets είναι μικρό και αραιό. Ως εκ τούτου, η ανάλυση θεμάτων που χρησιμοποιεί προσεγγίσεις με αναπαράσταση «συνόλου λέξεων» παραμένει ανεπαρκής για να αντικατοπτρίζει με ακρίβεια τα ενδιαφέροντα των χρηστών. Επιπλέον, οι τεχνικές συστάσεων που βασίζονται στο περιεχόμενο πρέπει να ισορροπούν μεταξύ της υπερβολικής σύστασης, δηλαδή της σύστασης πάρα πολλών στοιχείων, και της υπερβολικής εξειδίκευσης, δηλαδή της σύστασης στοιχείων που είναι πολύ παρόμοια με αυτά που έχουν ήδη προσπελαστεί και που περιέχουν επαναλαμβανόμενες πληροφορίες και δεν καλύπτουν το εύρος των ενδιαφερόντων του χρήστη.

Η μελέτη μας επικεντρώθηκε στην ανάπτυξη μιας νέας μεθόδου συστάσεων βάσει περιεχομένου [72] που χρησιμοποιεί γράφους γνώσης για (α) εξατομικευμένες προτάσεις tweet [71] και (β) εξατομικευμένες προτάσεις followee [69]. Αυτή η μέθοδος παρέχει ένα εναλλακτικό εξατομικευμένο χρονοδιάγραμμα που περιέχει tweets από την παγκόσμια ροή που ταιριάζουν έντονα με τα ενδιαφέροντα του χρήστη και εξατομικευμένες προτάσεις followees (¨ποιον να ακολουθήσω¨) με παρόμοια ενδιαφέροντα.

Η προσέγγισή μας χρησιμοποιεί τη σημασιολογική συνάφεια μεταξύ των ενδιαφερόντων των χρηστών και των θεμάτων των tweets ροής. Η επιλογή της σημασιολογικής συνάφειας ως κριτηρίου σύστασης έχει τα ακόλουθα πλεονεκτήματα. Πρώτον, λαμβάνοντας υπόψη ότι οι χρήστες διαβάζουν και γράφουν περιεχόμενο με ποικίλα θέματα, η μέθοδος σύστασής μας αξιοποιεί τις αντικειμενικές και αμετάβλητες συσχετίσεις μεταξύ αυτών των θεμάτων. Επιπλέον, η μέθοδός μας αντιμετωπίζει το πρόβλημα της υπερβολικής εξειδίκευσης αξιοποιώντας τον γράφο γνώσης για να προτείνει tweets σχετικών θεμάτων, και το πρόβλημα υπερβολικής σύστασης ταξινομώντας τα προφίλ χρηστών, δίνοντας έμφαση στα πιο συγκεκριμένα θέματα. Σε αντίθεση με τις τεχνικές συνεργατικού φιλτραρίσματος, η προσέγγισή μας δεν έρχεται αντιμέτωπη με το πρόβλημα της διαθεσιμότητας πόρων επειδή δεν χρησιμοποιεί τα δεδομένα του δικτύου των χρηστών. Τέλος, σε αντίθεση με τις μεθόδους «συνόλου λέξεων» (LDA, TF-IDF), η προσέγγισή μας αποφεύγει τα προβλήματα αποτελεσματικότητας που προκαλούνται από το περιορισμένο μέγεθος των tweets. Επιπλέον, εκχωρούμε ένα θέμα σε κάθε tweet ξεχωριστά σε αντίθεση με τις περισσότερες μεθόδους βάσει περιεχομένου, οι οποίες συγχωνεύουν τα tweets και χάνουν τον κατάλληλο βαθμό λεπτομέρειας για την εξαγωγή των θεμάτων.

Η προσέγγισή μας βασίζεται στην αναπαράσταση όλων των πιθανών ενδιαφερόντων των χρηστών ως έναν ιεραρχικό γράφο γνώσης, όπου κάθε κόμβος αντιστοιχεί σε ένα θέμα και οι ακμές υποδηλώνουν τη σχέση κατηγορίας-υποκατηγορίας μεταξύ των θεμάτων. Επιπλέον, το σύστημά μας βασίζεται στην αναπαράσταση του προφίλ οποιουδήποτε χρήστη ως ενός ταξινομημένου υπογράφου του γράφου γνώσης, έτσι ώστε οι κόμβοι να αντιπροσωπεύουν τα συγκεκριμένα ενδιαφέροντα των χρηστών. Και για τις δύο προτάσεις, tweet και followee, η μέθοδός μας χρησιμοποιεί το Δένδρο Steiner του προφίλ χρήστη για να υπολογίσει τη συνάφεια. Η διαίσθησή μας πίσω από τη χρήση του Δένδρου Steiner είναι ότι δεδομένου ενός συνόλου θεμάτων, το Δέντρο Steiner είναι ο υπογράφος του γράφου γνώσης που συνδέει με το λιγότερο κόστος τα θέματα ενδιαφέροντος (κόμβοι). Εάν μια έννοια συνδέει δύο ή περισσότερα θέματα ενδιαφέροντος στον γράφο γνώσης, τότε αυτή η έννοια είναι πιθανώς και η ίδια ένα θέμα ενδιαφέροντος. Αν και ορισμένα θέματα μπορεί να μην σχετίζονται άμεσα με τα ενδιαφέροντα του χρήστη, υποθέτουμε ότι εάν μια έννοια ανήκει στο δένδρο Steiner των θεμάτων ενδιαφέροντος, τότε η πιθανότητα να είναι θέμα ενδιαφέροντος και η ίδια αυξάνεται.

Β′.2.2 Αυτόματη δημιουργία αγνωστικών ως προς τα χαρακτηριστικά συνόλων δεδομένων για ανίχνευση ψευδών ειδήσεων στα μέσα κοινωνικής δικτύωσης

Οι αλγοριθμικές μέθοδοι για την αυτόματη ανίχνευση παραπλανητικού περιεχομένου στα μέσα κοινωνικής δικτύωσης είναι ζωτικής σημασίας ώστε να συμβάλουν στον μετριασμό της διάδοσης ψευδών ειδήσεων. Η ανάπτυξη μοντέλων ανίχνευσης μηχανικής μάθησης είναι μια διαδικασία που βασίζεται σε δεδομένα και εγείρει την ανάγκη για επαρκή και ποιοτικά σύνολα δεδομένων εκπαίδευσης, συμπεριλαμβανομένων δεδομένων από τα μέσα κοινωνικής δικτύωσης και των δεδομένων ground truth ψευδών ειδήσεων. Ωστόσο, υπάρχουν ορισμένα χαρακτηριστικά αυτού του προβλήματος που το καθιστούν μοναδικά δύσκολο.

- Η μη αυτόματη διαδικασία επισήμανσης περιεχομένου ειδήσεων και αναρτήσεων είναι χρονοβόρα και έχει ως αποτέλεσμα μερικά και ξεπερασμένα σύνολα δεδομένων.

- Υπάρχει σημαντική ποικιλομορφία μεταξύ των υπαρχόντων συνόλων δεδομένων που εγείρει σοβαρά προβλήματα στην αξιόπιστη αξιολόγηση και σύγκριση των μοντέλων ανίχνευσης.

- Οι περισσότερες προσεγγίσεις εστιάζουν σε συγκεκριμένα υποσύνολα χαρακτηριστικών που μπορούν να μετρηθούν με πολλούς τρόπους, ενώ τα χαρακτηριστικά διάδοσης δικτύου υποτιμώνται.

- Η φύση του περιεχομένου ψευδών ειδήσεων δεν είναι ομοιόμορφη και σχετίζεται με πρόσφατα αναδυόμενα, χρονικά κρίσιμα γεγονότα, τα οποία οι υπάρχουσες βάσεις γνώσεων και οι ελεγκτές γεγονότων ενδέχεται να μην έχουν επαληθεύσει ακόμη σωστά.

Για να αντιμετωπίσουμε αυτές τις προκλήσεις, αναπτύξαμε την υποδομή PHONY, η οποία αυτοματοποιεί τη δημιουργία συνόλων δεδομένων που περιέχουν ψευδείς ειδήσεις και τα ίχνη διάδοσής τους στο Twitter. Συγκεκριμένα, η υποδομή κάνει χρήση των ψευδών ειδήσεων που παρέχονται από υπηρεσίες ελέγχου γεγονότων και παράγει σύνολα δεδομένων που περιλαμβάνουν αναρτήσεις στα μέσα κοινωνικής δικτύωσης που αναφέρονται σε αυτές τις ψευδείς ειδήσεις. Αυτό επιτυγχάνεται με τη δημιουργία ενός αυξητικού αντεστραμμένου ευρετηρίου tweet που περιέχει tweets ροής. Αυτό το ευρετήριο αποτελείται από διευρυμένα tweets που περιλαμβάνουν το κείμενο, τα μεταδεδομένα και τα δεδομένα ιστού που σχετίζονται με κάθε tweet. Η δημιουργία ενός νέου συνόλου δεδομένων ξεκινά με τη συλλογή ψευδών ειδήσεων από ιστότοπους ελέγχου γεγονότων και στη συνέχεια με την ανάλυση και τη χρήση τους ως ερωτήματα στο ανεστραμμένο ευρετήριο.

Το PHONY δημιουργεί ομοιόμορφα, ευέλικτα και ενημερωμένα silver standard σύνολα δεδομένων, ενώ επιτρέπει στους χρήστες να επιλέξουν την πηγή ελέγχου γεγονότων, τη χρονική περίοδο και τον τύπο ψευδών ειδήσεων που τους ενδιαφέρουν, και η κεντρική ιδέα πίσω από την προσέγγισή μας περιγράφεται στο [102].

Τα σύνολα δεδομένων που δημιουργούνται είναι αγνωστικά ως προς τα χαρακτηριστικά, επομένως πρόκειται για σύνολα δεδομένων που δεν περιέχουν μετρήσεις συγκεκριμένων χαρακτηριστικών ταξινόμησης. Αυτό επιτρέπει στους χρήστες να επιλέγουν ελεύθερα τα χαρακτηριστικά και τις μετρήσεις που ταιριάζουν καλύτερα στις μεθόδους ταξινόμησης και ανίχνευσης που επιθυμούν. Τα σύνολα δεδομένων PHONY περιέχουν δεδομένα κειμένου, χρήστη, δικτύου και διάδοσης και, από όσο γνωρίζουμε, όλα τα χαρακτηριστικά που συναντάμε στη βιβλιογραφία μπορούν να εξαχθούν απευθείας χρησιμοποιώντας απλές ενέργειες.

Β′.2.3 Συστάσεις βασισμένες στη γνώση για εξερεύνηση δεδομένων

Τα τελευταία χρόνια, η αύξηση των δεδομένων, ο όγκος και η πολυπλοκότητα των διαθέσιμων δεδομένων έχουν αυξήσει την ανάγκη για αποτελεσματικά συστήματα εξερεύνησης δεδομένων. Τα συστήματα αναζήτησης συνήθως αντιμετωπίζουν τα ακόλουθα προβλήματα: ασάφεια φυσικής γλώσσας, αποτελεσματικότητα αναγνώρισης και σύνδεσης οντοτήτων, αναγνώριση οντοτήτων, λέξεων-κλειδιών και εννοιών που εξαρτώνται από τον θεματικό τομέα. Ταυτόχρονα, τα διαθέσιμα σύνολα δεδομένων πρέπει να είναι προσβάσιμα από καθημερινούς χρήστες που δεν είναι εξοικειωμένοι με τις βάσεις δεδομένων ή τον τύπο, τη δομή και το περιεχόμενο των δεδομένων.

Στο πλαίσιο της διατριβής, αναπτύξαμε το σύστημα KNOwDE που εστιάζει στην παροχή βοήθειας στους χρήστες που στοχεύουν στην εξαγωγή γνώσης από δεδομένα χωρίς να γνωρίζουν επακριβώς τη συγκεκριμένη δομή και το περιεχόμενο των δεδομένων. Το σύστημα αντιμετωπίζει το πρόβλημα της εξερεύνησης δεδομένων υπό το πρίσμα της δημιουργίας αποτελεσματικών συστάσεων βασισμένων στη γνώση και τα δεδομένα και της παροχής σχετικών πληροφοριών για τα δεδομένα στον χρήστη. Για να παρέχουμε συστάσεις για ερωτήματα σχετικά με τα ενδιαφέροντά τους και τα δεδομένα, χρησιμοποιούμε βάσεις γνώσεων και έναν γράφο που προέρχεται από τη βάση δεδομένων που είναι προς εξερεύνηση. Οι σημασιολογικά σχετικές συστάσεις εμπλουτίζονται χρησιμοποιώντας περαιτέρω συστάσεις αξιοποιώντας σχέσεις γενίκευσης και εξειδίκευσης από τις βάσεις γνώσης. Τέλος, το KNOwDE, χρησιμοποιώντας αλγόριθμους θεωρίας γράφων, προτείνει αντικείμενα δεδομένων και παρέχει δύο αναπαραστάσεις γράφων με βάση τις επιλογές των χρηστών, προσφέροντας μια οπτικοποίηση που επιτρέπει στους χρήστες να αποκτήσουν μια πιο ολοκληρωμένη οπτική στα αποτελέσματα και τους βοηθά να αλληλεπιδράσουν με τα προτεινόμενα αντικείμενα και τα αντικείμενα που σχετίζονται με τους.

Β′.2.4 Εφαρμογές που αξιοποιούν δεδομένα κοινωνικών δικτύων για εξερεύνηση συνόλων δεδομένων

Στο πλαίσιο της διατριβής, έχουμε αναπτύξει ορισμένες μεθόδους και υπηρεσίες, οι οποίες διευκολύνουν τους χρήστες που δεν έχουν ιδιαίτερες γνώσεις για τα δεδομένα να αξιολογήσουν, να πλοηγηθούν και να ανακαλύψουν χρήσιμες πληροφορίες στα διαθέσιμα σύνολα δεδομένων. Οι εφαρμογές αυτές είναι οι ακόλουθες:

1. **Υπολογισμός της προσοχής βάσει tweet για άρθρα σχετικά με την COVID-19**

   Το ξέσπασμα της πανδημίας του κορωνοϊού μας έστρεψε σε νέες προτεραιότητες όσον αφορά την ανάλυση των κοινωνικών δικτύων. Συγκεκριμένα, συνεργαστήκαμε με την ομάδα που αναπτύσσει το BIP!Finder[1] για να βοηθήσουμε στη δημιουργία ενός συνόλου δεδομένων που θα αποτελείται από βιβλιογραφία που σχετίζεται με την COVID-19 και θα περιλαμβάνει μια εναλλακτική μετρική, έναν δείκτη βασισμένο στην ανάλυση μέσων κοινωνικής δικτύωσης. Καθώς γράφουμε αυτές τις γραμμές, τα άρθρα σχετικά με την πανδημία δημοσιεύονται με ραγδαίο ρυθμό, γεγονός που καθιστά πολύ δύσκολη την αποτελεσματική εξερεύνηση και εξαγωγή χρήσιμων πληροφοριών από αυτά. Σε αυτό το πλαίσιο, ο κύριος στόχος της ομάδας ήταν να παράγει το BIP4COVID19, ένα ανοιχτά διαθέσιμο σύνολο δεδομένων που περιέχει διάφορες μετρικές που υπολογίζονται για τη βιβλιογραφία που σχετίζεται με την COVID-19 [145].

   Για να καταγράψουμε την δημοτικότητα των άρθρων, επιλέξαμε να μετρήσουμε την προσοχή που λαμβάνει κάθε άρθρο στο Twitter. Αυτή η εναλλακτική μετρική περιλαμ-

[1] https://bip.imsi.athenarc.gr/

βάνει τη μέτρηση του αριθμού των πρόσφατων tweets που αναφέρουν τα συγκεκριμένα επιστημονικά άρθρα. Θεωρούμε ότι αυτό είναι ένα μέτρο της προσοχής στα μέσα κοινωνικής δικτύωσης για κάθε βιβλιογραφικό άρθρο που σχετίζεται με την COVID-19.

Ωστόσο, η πρόσβαση σε ολόκληρη τη ροή των tweets ή η πραγματοποίηση πολλαπλών αναζητήσεων σε ιστορικά tweets θα έδινε ημιτελή αποτελέσματα λόγω των περιορισμών που θέτει το Twitter API. Έτσι, χρησιμοποιήσαμε ένα υπάρχον σύνολο δεδομένων με tweets που σχετίζονται με την COVID-19 και τα εξορύξαμε για διευθύνσεις URL που οδηγούν στα άρθρα της βάσης δεδομένων μας. Επομένως, δημιουργήσαμε τις διευθύνσεις URL των άρθρων στο doi.org, το PubMed και το PMC με βάση τα αντίστοιχα αναγνωριστικά. Παράγουμε περιοδικά τις διευθύνσεις URL για τη μέτρηση του δείκτη προσοχής στο Twitter με βάση τα πιο πρόσφατα tweets, τα οποία δημοσιεύουμε με συχνές ενημερώσεις εκδόσεων στο Zenodo [143] και στον ιστότοπο Bip4Covid[2].

2. **Συνδυασμός γεωχωρικών κοινωνικών δεδομένων για τη δημιουργία προφίλ αστικών περιοχών**

   Τα δεδομένα τοποθεσίας έχουν χρησιμοποιηθεί ευρέως για την ανίχνευση συμβάντων, την ανάλυση συναισθήματος, την αναγνώριση hotspot, την αναγνώριση τυπικών μοτίβων κίνησης, τον εμπλουτισμό χαρτών πόλεων κ.λπ. Ωστόσο, τα δεδομένα τοποθεσίας αντιμετωπίζουν ορισμένα σοβαρά προβλήματα:

   - Οι εθελοντικές γεωγραφικές πληροφορίες (VGI: εξάγονται από μέσα κοινωνικής δικτύωσης, πλατφόρμες microblogging, εφαρμογές check-in, GPS κινητών τηλεφώνων) που παρέχονται από διαδικτυακούς χρήστες είναι αρκετά ανακριβείς.
   - Οι πληροφορίες Σημείων Ενδιαφέροντος (PoI) (που προσφέρονται από κορυφαίους ομίλους όπως η Google, Here, Bing, Foursquare) θέτουν περιορισμούς στη χρήση αυτών των API, παρέχοντας έτσι στους χρήστες μια πολύ περιορισμένη προβολή της υπάρχουσας υποδομής μιας πόλης που δεν μπορεί να χρησιμοποιηθεί απευθείας για την εξαγωγή πρόσθετων πληροφοριών για περιοχές ολόκληρης πόλης.
   - Δεδομένα από κρατικούς φορείς, παρόλο που είναι επίσημα, επιμελημένα, εξαιρετικής ποιότητας και αδύνατο να συλλεχθούν από ιδιώτες, έχουν το προφανές μειονέκτημα ότι δεν μπορούν να παρέχονται σε πραγματικό χρόνο, συνήθως δεν είναι διαθέσιμα μέσω API και το πιο σημαντικό, μπορεί να ενημερώνονται σε πολύ σπάνια διαστήματα (π.χ. δεδομένα απογραφής), επομένως υπάρχει κίνδυνος να είναι μάλλον ξεπερασμένα.

   Για να βοηθήσουμε τους καθημερινούς χρήστες να εξερευνήσουν διαθέσιμα δεδομένα που σχετίζονται με αστικές περιοχές, αναπτύξαμε την υποδομή CitySense. Το CitySense είναι ένα δυναμικό πρόγραμμα προβολής αστικών περιοχών που ενσωματώνει διάφορα σύνολα δεδομένων που σχετίζονται με μια αστική περιοχή, παρέχοντας μια πλούσια απεικόνιση της ζωής της πόλης. Η εργασία μας επικεντρώθηκε στο συνδυασμό διαφορετικών συνόλων δεδομένων διαφόρων προελεύσεων για να παρέχει μια πιο ολοκληρωμένη εικόνα μιας γεωγραφικής περιοχής. Για να το πετύχουμε αυτό, αναπτύξαμε το CityProfiler, ένα υποσύστημα υπεύθυνο για τη συλλογή δεδομένων, και δημιουργήσαμε τη βάση δεδομένων CitySense που αποθηκεύει τα ποικίλα δεδομένα. Για να εμπλουτίσουμε τα δεδομένα Σημείων Ενδιαφέροντος, σχεδιάσαμε το CityProfiler για να συλλέγει γεωχωρική δραστηριότητα στο Twitter [70] που δημιουργείται από τους χρήστες. Επιπλέον, εστιάσαμε στην αποτελεσματική χωρική συγκέντρωση, οπτικοποίηση και παρουσίαση στον τελικό χρήστη με μια πλούσια απεικόνιση οποιασδήποτε πηγής δεδομένων, έτσι ώστε οι χρήστες να μπορούν να ερμηνεύουν εύκολα αυτές

[2]https://bip.covid19.athenarc.gr/

τις πληροφορίες. Κάναμε το CitySense διαθέσιμο στους χρήστες μέσω της αντίστοιχης διαδικτυακής εφαρμογής[3] που δημιουργεί μια ενοποιημένη εικόνα της περίπτωσης χρήσης μας, την αστική περιοχή του Chicago, χρησιμοποιώντας ανοιχτά δεδομένα από διοικητικές πηγές, διαδικτυακά API PoI και tweets.

Β′.3 Συνεισφορές διατριβής

- **Εξατομικευμένες συστάσεις tweet και followee με βάση γράφους γνώσης**

  Αναπτύξαμε μια νέα μέθοδο βασισμένη στο περιεχόμενο που χρησιμοποιεί γράφους γνώσης για (α) εξατομικευμένες προτάσεις tweet και (β) εξατομικευμένες προτάσεις followee. Οι συστάσεις tweet συνθέτουν ένα εναλλακτικό εξατομικευμένο χρονοδιάγραμμα που περιέχει tweets ροής που ταιριάζουν έντονα με τα ενδιαφέροντα του χρήστη και εξατομικευμένες προτάσεις followee που είναι μια ταξινομημένη λίστα λογαριασμών Twitter με παρόμοια ενδιαφέροντα με τον χρήστη. Και οι δύο μέθοδοι μας:

  1. βασίζονται στην αναπαράσταση των προφίλ χρηστών ως θέματα ενδιαφέροντος. Στο πλαίσιο της εργασίας μας, τα θέματα ενδιαφέροντος είναι κόμβοι ενός προκαθορισμένου γράφου γνώσης που αντιπροσωπεύουν τα ενδιαφέροντα συγκεκριμένων χρηστών και που συνδέονται με τρόπο χαμηλού κόστους χρησιμοποιώντας τον αλγόριθμο Steiner Tree
  2. μπορεί να προσαρμοστεί για να καλύψει νέα θέματα ενδιαφέροντος και να μειώσει τις επιπτώσεις της υπερβολικής εξειδίκευσης και της υπερβολικής σύστασης
  3. δεν επηρεάζονται από τους περιορισμούς που θέτει το Twitter σχετικά με τη διαθεσιμότητα των δεδομένων δικτύου

  Πραγματοποιήσαμε δύο πειράματα: ένα για να αξιολογήσουμε το σύστημα συστάσεων για tweet και followee και ένα για το οποίο χρησιμοποιήσαμε ένα μεγάλο σύνολο δεδομένων για να αξιολογήσουμε την αποτελεσματικότητα και την κλιμάκωση της προσέγγισής μας. Η αποτελεσματικότητα της μεθόδου μας ξεπερνά σε πολλές περιπτώσεις τις προσεγγίσεις της τεχνολογικής στάθμης, τις οποίες υλοποιήσαμε για τους σκοπούς της αξιολόγησης, και αποφέρει καλά αποτελέσματα όσον αφορά την ακρίβεια και την κλιμάκωση στο χρόνο.

- **Αυτόματη δημιουργία αγνωστικών χαρακτηριστικών συνόλων δεδομένων για την ανίχνευση ψευδών ειδήσεων στα μέσα κοινωνικής δικτύωσης**

  Αναπτύξαμε το PHONY, μια υποδομή για την αυτοματοποίηση της δημιουργίας silver standard συνόλων δεδομένων αγνωστικών χαρακτηριστικών. Αυτά τα σύνολα δεδομένων περιέχουν ψευδείς ειδήσεις και τα ίχνη της διάδοσής τους στο δίκτυο του Twitter με βάση τις ψευδείς ειδήσεις που παρέχονται από ιστότοπους ελέγχου γεγονότων και περιλαμβάνουν τα απαραίτητα δεδομένα για την εξαγωγή όλων των χαρακτηριστικών που συναντώνται στη βιβλιογραφία, συμπεριλαμβανομένης της διάδοσης στο δίκτυο και των σημασιολογικών χαρακτηριστικών.

  Επιπλέον, διερευνήσαμε το εύρος των χαρακτηριστικών ανίχνευσης παραπληροφόρησης που συναντώνται στη βιβλιογραφία. Η διατριβή παρέχει μια πλήρη τυπολογία χαρακτηριστικών που βασίζεται στην ανάλυση και συστηματοποίηση όλων των διαθέσιμων χαρα-

[3] http://geoprofiler.imsi.athenarc.gr/

κτηριστικών. Αυτή η τυπολογία μπορεί να είναι χρήσιμη για την εκπαίδευση μοντέλων ανίχνευσης ψευδών ειδήσεων που καλύπτουν όλους τους τύπους παραπληροφόρησης.

Επιπλέον, χρησιμοποιήσαμε το PHONY για να δημιουργήσουμε το ελληνικό σύνολο δεδομένων PHONY, ένα μεγάλο σύνολο δεδομένων ψευδών ειδήσεων που διαδίδονται στην ελληνική σφαίρα του Twitter. Η αποτελεσματικότητα του PHONY μετρήθηκε με αξιολόγηση του ελληνικού συνόλου δεδομένων PHONY που έφτασε μία μέση ακρίβεια του 77,5

- **Συστάσεις βασισμένες στη γνώση για εξερεύνηση δεδομένων**

  Αναπτύξαμε το KNOwDE, ένα σύστημα για τη δημιουργία αποτελεσματικών συστάσεων βασισμένων στη γνώση και σε δεδομένα για την εξερεύνηση δεδομένων βάσει ερωτημάτων αναζήτησης χρηστών. Η μέθοδος πίσω από το KNOwDE είναι ότι οι έννοιες στο ερώτημα του χρήστη αντιστοιχίζονται με τις έννοιες στη βάση δεδομένων με τη βοήθεια βάσεων γνώσεων που παρέχουν εναλλακτικές λέξεις-κλειδιά και φράσεις-κλειδιά που σχετίζονται σημασιολογικά με τις αρχικές, για να καταγράψουν το πραγματικό εννοιολογικό πλαίσιο των ερωτημάτων των χρηστών. Συγκεκριμένα, χρησιμοποιούμε προτάσεις που βασίζονται σε βάσεις γνώσεων και έναν γράφο που εξάγεται από τη βάση δεδομένων (γράφος δεδομένων) για να βοηθήσουμε τους χρήστες που δεν είναι σίγουροι πώς να σχηματίσουν ένα σωστό ερώτημα. Αυτές οι προτάσεις βοηθούν τους χρήστες να σχηματίσουν ερωτήματα σχετικά με τα ενδιαφέροντά τους και τη βάση δεδομένων. Ο χρήστης μπορεί να αλληλεπιδράσει με το σύστημα επεκτείνοντας το αρχικό ερώτημα με αυτές τις εναλλακτικές λέξεις-κλειδιά που ταξινομούνται αναλύοντας γράφους λέξεων-κλειδιών, δηλαδή αναπαραστάσεων για κάθε λέξη-κλειδί και σχετικές εναλλακτικές. Επιπλέον, το KNOwDE επεκτείνει αυτές τις εναλλακτικές λύσεις και προσφέρει περαιτέρω συστάσεις βασισμένες σε σχέσεις γενίκευσης και εξειδίκευσης που προέρχονται από τις βάσεις γνώσεων.

  Στη συνέχεια, το KNOwDE χρησιμοποιεί τις επιλεγμένες λέξεις-κλειδιά και έναν γράφο που προέρχεται από τη βάση δεδομένων (γράφος δεδομένων) για να δημιουργήσει συστάσεις ταξινομημένων αντικειμένων από τα δεδομένα με βάση τη συχνότητά τους και την PageRank βαθμολογία τους. Για να παρέχουμε μια πιο ολοκληρωμένη ματιά στα αποτελέσματα, δημιουργούμε δύο προβολές του γράφου δεδομένων με βάση τις επιλεγμένες εναλλακτικές λέξεις-κλειδιά, δηλαδή τον Υπογράφο S και το δένδρο Steiner, που συνδέουν τα πιο κρίσιμα και σχετικά αντικείμενα της βάσης δεδομένων.

  Έχουμε υλοποιήσει το KNOwDE, το οποίο επί του παρόντος ενσωματώνει τη βάση δεδομένων CORDIS[4] και τις βάσεις γνώσεων DBPedia[5] και ConceptNet[6]. Έχουμε επίσης αναπτύξει την εφαρμογή KNOwDE[7] χρησιμοποιώντας μια φιλική προς το χρήστη διεπαφή.

- **Υπολογισμός της προσοχής βάσει Tweet για άρθρα σχετικά με την COVID-19**

  Σχεδιάσαμε μια μεθοδολογία για τον υπολογισμό ενός εναλλακτικού δείκτη προσοχής που εξάγεται από δεδομένα κοινωνικών δικτύων για την κατάταξη της βιβλιογραφίας που σχετίζεται με την COVID-19. Για το σκοπό αυτό, εξορύξαμε ένα υπάρχον σύνολο δεδομένων με tweets που σχετίζονται με την COVID-19 για ένα σύνολο διευθύνσεων URL που παραπέμπουν στα άρθρα που σχετίζονται με την COVID-19. Αυτός ο δείκτης, που ονομάζεται Προσοχή Social Media, ήταν η συνεισφορά μας στο σύνολο δεδομένων

[4]https://cordis.europa.eu/en
[5]https://wiki.dbpedia.org/
[6]https://conceptnet.io/
[7]http://knowde.imsi.athenarc.gr/

Bip4COVID[8]. Το BIP4COVID19, είναι ένα ανοιχτό διαθέσιμο σύνολο δεδομένων που περιέχει διάφορες μετρικές για τον αντικτύπου που υπολογίζονται για τη βιβλιογραφία που σχετίζεται με την COVID-19 και το οποίο έχει ήδη συγκεντρώσει μεγάλη προσοχή στο Zenodo (151.570 μοναδικές προβολές και 14.232 μοναδικές λήψεις μέχρι στιγμής).

- **Συνδυασμός γεωχωρικών κοινωνικών δεδομένων για δημιουργία προφίλ αστικών περιοχών**

  Έχουμε σχεδιάσει και αναπτύξει το πλαίσιο CitySense, ένα δυναμικό πρόγραμμα προβολής αστικών περιοχών που ενσωματώνει διάφορα σύνολα δεδομένων που σχετίζονται με μια αστική περιοχή, παρέχοντας μια πλούσια απεικόνιση της ζωής μιας πόλης. Το CitySense συνδυάζει δεδομένα από διοικητικές πηγές, API σημείων ενδιαφέροντος και το κοινωνικό δίκτυο Twitter για να παρέχει μια ενοποιημένη προβολή όλων των διαθέσιμων χωρικών πληροφοριών για μια συγκεκριμένη αστική περιοχή. Για το σκοπό αυτό, αναπτύξαμε το CityProfiler, ένα υποσύστημα υπεύθυνο για τη συλλογή δεδομένων και την αποτελεσματική χωρική συγκέντρωση, και τη βάση δεδομένων CitySense που αποθηκεύει τα δεδομένα που συλλέγονται.

  Επιπλέον, αναπτύξαμε την αντίστοιχη διαδικτυακή εφαρμογή [9] που δημιουργεί μια ενοποιημένη άποψη της αστικής περιοχής του Σικάγο. Σε αυτό το πλαίσιο, εστιάσαμε στην οπτικοποίηση και παρουσίαση στον τελικό χρήστη με μια πλούσια προβολή των διαθέσιμων δεδομένων για την εύκολη ερμηνεία αυτών των πληροφοριών μέσω μιας φιλικής διεπαφής χρήστη.

[8]https://zenodo.org/record/5560080
[9]http://geoprofiler.imsi.athenarc.gr/

Appendix C

Glossary

Γλωσσάρι

Μετάφραση	Αγγλικός Όρος
k-σημαντικότερα	top-k
αγνωστικό ως προς τα χαρακτηριστικά	feature-agnostic
ανάκτηση πληροφορίας	information retrieval
ανάλυση θεμάτων	topic analysis
αναπαράσταση «συνόλου λέξεων»	bag-of-words representation
αναφορά	reference
αντεστραμμένο ευρετήριο	inverted index
αντικείμενο δεδομένων	data object
αντικείμενο του διαδικτύου	web object
απήχηση	impact
αφοσίωση των χρηστών	user engagement
βαθμολογία	score
βαθμός λεπτομέρειας	granularity
βάση γνώσης	knowledge base
γράφος γνώσης	knowledge graph
δεδομένα δικτύου χρηστών	user network data
δένδρο Steiner	Steiner tree
δημιουργία προφίλ χρήστη	user profiling
δημοφιλία	popularity
διάδοση των ψευδών ειδήσεων	fake news diffusion
διαχείριση δεδομένων και γνώσης	data and knowledge management
διευρυμένα τωεετς	broadened tweets
δίκτυο αναφορών	citation network
εναλλακτική μετρική	altmetric
εξατομίκευση	personalization
εξερεύνηση δεδομένων βάσει γνώσης	knowledge-based data discovery
εξερεύνηση δεδομένων	data exploration
εξόρυξη κειμένου	text mining
επισήμανση περιεχομένου	content labelling
επιστημονικές δημοσιεύσεις	scientific publications
επιτήρηση μαζών	mass surveilance

θέμα ενδιαφέροντος	topic of interest
ιεραρχικός γράφος	hierarchical graph
ισχυρά συνδεδεμένος γράφος	strongly connected graph
ίχνος διάδοσης	diffusion footprint
κατάταξη	ranking
κεντρικότητα	centrality
κλειστότητα	closeness
κλιμάκωση	scalability
λανθάνον θέμα	latent topic
λανθάνουσας ανάθεσης Dirichlet	latent dirichlet allocation (LDA)
μέθοδοι κατάταξης	ranking methods
μεταδεδομένα	Metadata
μηχανική μάθηση	machine learning
μηχανισμός προσοχής	attention mechanism
μοντέλο ανίχνευσης	detection model
μοντελοποίηση θεμάτων	topic modelling
ομάδα-στόχος	target-group
οργανισμοί ειδήσεων	news organizations
πρόβλημα ψυχρής εκκίνησης	cold-start problem
προγραμματιστική διεπαφή εφαρμογών	application programming interface (API)
προσοχής βάσει τωεετ	tweet-based attention
προτύπων διάχυσης	diffusion patterns
ροή τωεετς	tweet stream
ερώτημα	query
σημασιολογικά χαρακτηριστικά	semantic features
σημασιολογική συνάφεια	semantic relevance
σημασιολογική σχέση	semantic relation
σημασιολογικού περιεχομένου	semantic content
συνολική επίδραση	influence
σύνολο δεδομένων	dataset
σύνολο δεδομένων εκπαίδευσης	training dataset
συστάσεις που βασίζονται στο περιεχόμενο	content-based recommendations
σύστημα συστάσεων	recommender
ταξινομητής	classifier
τιμή βάσης	default
τυχαία περιήγηση	random walk
υπερβολική εξειδίκευση	over-specialization
υπερβολική σύσταση	over-recommendation
υπόβαθρο αληθείας	ground truth
φαινόμενα παραπλάνησης	misleading phenomena
φαινομενα παραπληροφόρησης	misinformation phenomena
χαρακτηριστικό ανιχνευσης	detection feature
χαρακτηριστικό διάδοσης	diffusion feature
χρονοδιαγράμμα	timeline
ψευδείς ειδήσεις	fake news