Temporal Word Embeddings for Dynamic User Profiling in Twitter

by

Breandán Kerin, B.A., B.A.I.

A Dissertation submitted to the University of Dublin, Trinity College

in fulfilment of the requirements for the degree of

Integrated Masters in Computer Engineering

Supervised by Dr. Séamus Lawless

Co-Supervised by Dr. Annalina Caputo

Submitted to the University of Dublin, Trinity College

April 2019

Declaration

I, Breandán Kerin, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

_____ _____

Breandán Kerin 12th April 2019

Summary

The research described in this dissertation is focused on exploring the domain of user profiling. As defined by the Oxford English dictionary, a user profile is “a collection of information or data about the habits, preferences, etc., of a user, especially of a product or service”. [1] User profiling is a relatively nascent technology and research domain, which is hugely important in providing positive, personalised experiences to application users by tailoring content to their interests, attributes, moral values and lifestyle.

In research, it is clear that the variation of user characteristics through time is a problem of significant interest. However, upon conducting an extensive review of user profiling research literature it was found that there has been limited research conducted to-date into how temporal aspects of users can be captured using user profiling techniques. Coupled with a notable lack of research into the use of embedding techniques to capture temporal variances in both language and entities in the same vector space, a research question was formulated which sought to fill these gaps and hence enhance the state of the art in temporal user profiling.

This research saw the development of an end-to-end temporal user profiling system, which built upon the well-known Temporal Random Indexing word embedding technique to enable the interests of Twitter users to be captured through analysis of their use of language. The resultant user embeddings were evaluated against the state-of-the-art technique in this domain, where the implemented system was found to outperform it in the generation of temporal user profiles.

Despite the fact that the findings of the research were limited by unanticipated setbacks, this research has seen the development of a novel temporal user profiling system, capable of generating temporal user profiles for short streams of text through the use of Temporal Random Indexing. The fact that a Twitter user data can be processed, enriched, and used to produce both user and word embeddings in a single application is a significant accomplishment of this research, and a meaningful contribution to the knowledge of the research community in the domain of user profiling.

Acknowledgements

This MAI research project has been a tough, gruelling and significant undertaking that I could not have accomplished without the amazing support and guidance of so many different people that I have been lucky to meet on this incredible journey.

First and foremost my thanks goes to Dr. Séamus Lawless and Dr. Annalina Caputo, both of who have been an incredible support to me through all the trials and tribulations of this project. Their insights, patience, guidance and support have been second-to-none.

I am incredibly fortunate to have a wonderful support network: To Manjot, Martin, Eileen, Romy (#Bestie) and Holly and Sinéad, thank you for listening to all of my ranting and raving and for being there to offer your advice, opinions and help with my issues (whether technical or motivational)!

Lastly, to my loving family - Mom, Dad, Thomas, Seán - and my incredible partner, Amber. You have all invested so much in support of my dreams and aspirations, and I could never have accomplished any of this without you.

BREANDÁN KERIN

University of Dublin, Trinity College,

April 2019

Abstract

The research conducted under this title focused on exploring the domain of user profiling, a nascent technology which has been steadily attracting increased interest from the research community as its potential in the provision of personalised digital services is realised.

An extensive review of related literature revealed that there has been limited research conducted to-date into how temporal aspects of users can be captured using user profiling techniques. Coupled with the notable lack of research into the use of word embedding techniques to capture temporal variances in language, an opportunity was identified to extend the Random Indexing word embedding technique such that it could model the interests of users based on their use of language.

To achieve this, the work completed concerned itself with extending an existing implementation of Temporal Random Indexing to model Twitter users across multiple granularities of time. The product of this was a novel approach to producing a set of vectors describing the evolution of each Twitter user's interests over time through their use of language. These vectors were evaluated against another state-of-the-art word embedding technique, the Word2Vec Skip-gram model, where it was found that Temporal Random Indexing outperformed Word2Vec in the generation of temporal user profiles.

The major contribution of this research has been the development of a novel temporal user profiling system, capable of generating temporal user profiles for short streams of text through the use of Temporal Random Indexing. The fact that Twitter user data can be processed, enriched, and used to produce both user and word embeddings in a single application is a significant accomplishment of this research, and a meaningful contribution to the knowledge of the research community in the domain of user profiling.

Table of Contents

List of Tables

List of Figures

List of Code Snippets

List of Equations

Chapter 1 Introduction

This research explores the potential of temporal word embeddings in generating temporal user profiles, also commonly referred to as temporal user models. Such temporal embedding methods are capable of capturing variance in word usage through time as the language of the corpus varies. In particular, this research examines the Temporal Random Indexing (**TRI**) embedding method for its effectiveness in the generation of such profiles. The research described in this dissertation evaluates TRI against the state-of-the-art in temporal user profiling techniques.

Section 1.1 gives an overview of the domains upon which this research is based, outlining the context of the problem domain and enumerating some of the open challenges in the domain of user profiling.

Section 1.2 defines the research question and outlines the objectives that are addressed in this research, motivated by the discussion provided in *Section 1.1*.

Section 1.3 highlights the contributions made by this research to the domain of user profiling.

Finally, *Section 1.4* details the structure and content of the remainder of this report.

1.1 Problem Area

As of the time of writing, it is estimated that approximately 4.2 billion people of an estimated 7.6 billion globally are connected to the internet in some way or another[1]. This enormous population of internet users is matched by an estimate of over 1.9 billion live websites[2]. Some of the most successful live websites in 2019 are social networking sites, hosting platforms such as Twitter, Facebook and YouTube which allow users to connect and share online content with each other. In order to capture, maintain and even increase the engagement of such a large user base in such a complex environment, the organisations behind these platforms are increasingly employing personalisation tactics, where the preferences and interests of the user are modelled,

[1] Estimate by Internet Live Stats, a widely cited website whose statistics are generated using the Worldometer RTS algorithm on data collected by multiple sources including the ITU, UN and others. [82]
[2] Estimate by Internet Live Stats. [83]

clustered and learned in order to deliver tailored content such as advertisements and articles directly to them.

User profiling is a data analytics approach commonly used to support platform personalisation. As explained by Kanoje *et al.*, it is "the process of identifying the data about a user interest domain. This information can be used by the system to understand more about [the] user and this knowledge can be further used for enhancing the retrieval for providing satisfaction to the user." [1] Considering this dramatic change in how users experience and interact with content, it is clear why internet giants such as YouTube, Facebook and Twitter are investing heavily in gaining a better understanding of their users: Resulting in the global domain of big data analytics being worth billions of dollars in 2019[3].

User profiling is a contentious technology from an ethical perspective: Whether organisations use it in accordance with legal and ethical guidelines is highly debatable. Several multinational technology companies have come under fire for leveraging and capitalising on users' data without obtaining their explicit consent and knowledge.

- In January 2019, Google was fined approximately €50 million for "not properly disclosing to users how data is collected across its services — including its search engine, Google Maps and YouTube — to present personalized advertisements"[4].
- In July 2018, Facebook were fined £500,000 by UK regulators because they "failed to provide the kind of protections they are required to under the Data Protection Act"[5] when Cambridge Analytica, a company hired by then-presidential candidate Donald Trump, used Facebook user data in order to "identify the personalities of American voters and influence their behavior"[6].

Despite the nefarious actions of some organisations, it is the opinion of the author that user profiling has huge potential and desirability from the perspective of improving user experience. Though a relatively nascent technology, user profiling can be applied in simplifying navigation

[3] Statista, "Forecast of Big Data market size, based on revenue, from 2011 to 2027 (in billion U.S. dollars)" (2019). Available: [84]

[4] Article by the New York Times, which analyses the penalty applied to Google "Under Europe's Data Privacy Law". [87]

[5] Article by the Guardian (UK), which analyses the penalty levied against Facebook for "lack of transparency and failing to protect users' information". [85]

[6] An article published by the New York Times, offering an explanation of "Facebook and Cambridge Analytica: What You Need to Know as Fallout Widens". [86]

of the internet and all of its complex components through personalisation, allowing relevant content to be delivered to users more efficiently.

The idea of temporally modelling users can be motivated by the fact that an individual user and their data are not static: Their interests and preferences evolve and vary through time, often following patterns such as trends, periodicities and spikes. This was demonstrated by Arielle *et al.* (2013), who found that an individual's musical interests vary through time, and tend to fluctuate and change around "particular life changes" of the individual. [2] Their findings illustrate the inherent variability of user interests and preferences over time: Hence, it is clear that strong user profiling techniques should capture these variances.

The idea of capturing temporal variances whilst modelling users and their interests through time has not been the subject of a great deal of research at the time of writing of this document. Thus, exploring new viable approaches to capturing temporal variations in user profiling tasks is the primary motivation behind this research.

1.2 Research Objectives

The research question addressed by this research is as follows:

To what extent can a user's vocabulary on an online social network be used to infer their interests as they vary through time, using a previously unexplored word embedding technique?

This question was devised based on a comprehensive review of the state-of-the-art in the domain of user profiling, which highlighted (i) the ability of user profiling to provide enhanced user experiences, (ii) the lack of existing research into capturing temporal variations in user profiling models, and (iii) the potential of existing word embedding techniques to capture temporal variances in language use. These considerations are discussed in depth in *Chapter 2* and *Chapter 3* of this document.

The approach to evaluating the research with respect to this research question is dependent on (i) the word embedding technique employed, and (ii) the dataset used. However, any evaluation will certainly involve the comparison of produced results to a set of ground truth data, allowing for performance metrics to be obtained and assessed with respect to existing state-of-the-art approaches. During the Design phase of this research, a set of objectives are set out. These objectives are discussed in detail in Section 3.1.2.

1.3 Contributions of Research

The final contribution of this research is a fully operational dynamic user profiling solution, which models users based on short streams of text. The specific outputs are as follows:

1. A temporal user profiling system, which can construct temporal models of Twitter[7] users by applying the TRI word embedding technique to a Twitter dataset[8];
2. An evaluation environment which can be used to evaluate the output of the temporal user profiling system against another state of the art temporal user profiling technique.

This research lays the foundations for further work to be carried out in the development of effective user profiling techniques: In particular, with respect to the use of word embedding techniques.

1.4 Report Structure and Contents

The remainder of this dissertation is structured as follows.

Chapter 2 is entitled State of the Art, and provides an in-depth overview of the domain of user profiling, motivating its importance in modern intelligent systems as well as its significance and implications for users of such systems that employ them. Next, temporal user profiling is explored, with a review of significant research in this area to-date. Finally, a review of techniques for representing user data and models is conducted.

Chapter 3 is entitled Design. It first describes how the research question was formulated and the resulting research objectives, before detailing the design considerations and challenges that were involved in the development of the solution. These considerations include implementation decisions, evaluation considerations and an assessment of ethical implications of the research.

Chapter 4, entitled Implementation, describes the implementation phase of the project in great detail. In particular, this focuses on describing the steps involved in developing the various components of the temporal user profiling system, and the processing performed on the Twitter dataset.

[7] The motivations for choosing the Twitter platform for analysis are detailed in later chapters.
[8] In this system, both Twitter users and their vocabulary are modelled in the same vector space as a function of time for different time granularities using TRI.

Chapter 5 is the Evaluation chapter. It provides a quantitative evaluation of the implemented system and details the experiments that were carried out. The results are then discussed.

Finally, *Chapter 6* is the Conclusions and Future Works chapter. It gives some overall conclusions regarding the research as a whole, as well as potential avenues for the continuation and enhancement of the work completed in this research. It closes with some final remarks based on the research conducted.

Chapter 2 State of the Art

This chapter provides a review of the state of the art in user profiling in 2019, with the aim of providing the reader with a comprehensive understanding of this research domain.

Section 2.1 gives an overview of the domain of user profiling in 2019. This section provides an overview of the applications of user profiling developed in recent years as well as brief discussion into the area of data privacy with regards to user profiling.

Section 2.2 describes the applications of user profiling in OSNs and examines a number of closely related projects encountered during this research, highlighting their interesting attributes and contributions to the domain of user profiling.

Section 2.3 describes several different ways of representing user. This section focuses in particular on the area of word and temporal word embedding methods for modelling users data.

Finally, *Section 2.4* summarises the knowledge gained from the state of the art as explored in this chapter.

2.1 User Profiling

In 2019, the world is increasingly becoming driven by data. Digital data is being generated, updated and consumed at a rate never seen before in history. It is estimated that the social media giant Twitter stores and processes approximately 8TB of data per day, which translates into approximately 250 million tweets in a single 24-hour period. [3] Taken in the wider context of the scale of the internet, it is hard to quantify just how much data is being generated by the 1.9 billion live websites on the internet in a single day. [4]

Matching this massive scale of data generation is the volume of research being carried out into leveraging it. The domain of data analytics is continuing to grow and metamorphosise at breakneck speed as major players across all industries are discovering how it can be used to make their operations more efficient, effective and valuable. [5] Such research is being undertaken by both academics and industry professionals, with organisations commissioning specialised research groups to tackle some of their biggest business problems with data-driven solutions. One of the areas which shows great potential in this regard is that of user profiling.

2.1.1 Overview

User profiling, also commonly referred to as user modelling, is a well-established sub-field of Artificial Intelligence (**AI**): There has even been an annual conference established which is dedicated entirely to research work completed in the domain of user profiling. [6] The Oxford English dictionary gives two definitions for the term *user profile*: [7]

1. *User profile:* Computing the unique configurations, preferences, settings, etc., set up for or by a computer user, especially as stored on a server and accessible via various network computers.
2. *User profile*: A collection of information or data about the habits, preferences, etc., of a user, especially of a product or service.

Both of these definitions emphasise the derivation of an understanding about a user and their preferences based on information obtained about them. In simple terms, user profiling can be described as a process of acquiring data[9], performing any required processing on it, and manipulating it to produce a comprehensive model or representation of a user or group of users.

In general, there are two primary classifications of user profiles, as follows:

1. **Static versus Dynamic Models**
   User profiling models are generally either static or dynamic.
   - Static models are the most basic type of user profiling model. The modelling data captured in this case is treated as static and is not updated based on any shifts in the user's data.
   - Dynamic models are more flexible than their static counterparts, incorporating updates to the user's data as part of the model, accounting for changes in their interactions with the system in question.
2. **Empirical versus Analytical Models**
   User profiling models are usually also classified as being either empirical or analytical.
   - Empirical models, as the name suggests, are constructed based on empirical observations about the user rather than simulating cognitive processes. A

[9] The type of user data could vary from system settings to written text such as that found on a microblogging website like Twitter, to a user's connections and the geo-location of the content they generate.

common example of an empirical model is stereotype modelling, where every user is mapped to a *stereotype* i.e. a group of users with similar characteristics.

- Analytical profiling models on the other hand attempt to simulate cognitive processes, and do not try to model users based on shared characteristics with other users. These simulations are often based on underlying assumptions that are made about the users.

When it is considered that user profiling techniques can be applied by companies to model the consumers that use their products and services, the reason for the growing relevance of this research domain becomes clear.

2.1.2 Applications

The primary applications of interest for user profiling are those where consumer data is involved, since organisations have a deep interest in inferring additional insights from it for the purposes of monetisation and targeted advertising. The leaders in the field of user profiling research are Online Social Networks (**OSN**s), who are uniquely positioned to infer such insights.

2.1.2.1 Online Social Networks

OSNs are the communication status quo in 2019 for many demographic groups across the world, with hundreds of millions of active users using the instant messaging, content sharing and information seeking capabilities that they provide. Such widespread adoption means that popular OSNs such as Facebook, Instagram, Twitter, LinkedIn and YouTube must process gargantuan quantities of related data to provide their services. Whilst providing the resources to handle this volume of data is both expensive and an enormous operational challenge, the processing of this data gives OSNs a significant advantage over their non-OSN counterparts: The ability to obtain rich insights about their users through user profiling. Their native access to detailed, real-time user interaction information enables them to develop well-constructed and thoroughly trained user profiling models.

Enriched consumer insights are highly desired by any consumer-driven organisation. Facebook, the world's biggest OSN with approximately 2.32 billion users as of Q4 2018[10],

[10] The statistics provided here were obtained from Statista. [73] (Accessed 16/03/2019)

generates much of its revenue from selling consumer insights to organisations which do not have access to such insights themselves. Organisations armed with such insights are capable of providing services which are more closely tailored to the wants and needs of consumers, by using more targeted and relevant advertising. The research described in this document is based on data from users of the Twitter OSN, the motivation for which is made clear in *Section 3.2.1.1*.

2.1.2.2 User Profiling in Online Social Networks

Though every OSN has a unique approach to doing so, each offers a platform for connecting with other users and sharing content with them in a simple format with easy-to-use options. The combined power and simplicity is a strong incentive for users to use these platforms. The result for the platform operators is that their users cumulatively generate huge quantities of personal data which can be used to better understand their behaviours, needs, preferences and interests through profiling.

User profiling using OSN data has been the subject of much research. A review of related user profiling literature is described below.

2.1.2.2.1 Personality Inference

Personality inference using OSN data is a research problem which has been tackled by multiple researchers.

- D. Quercia *et al.* (2011) proposed a method for analysing a Twitter user's tweets to infer their OCEAN[11] personality traits. [8] The data for their study was collected using the Twitter API, and was limited to the few hundred Twitter users who shared their personality score from a Facebook application called *myPersonality*.
- R. Wald *et al.* (2012) proposed a method for inferring user personality traits based on Facebook data, [9] by using demographic and text-based attributes extracted from Facebook user profiles. Interestingly, the researchers make reference to the privacy implications of their research in terms of "allowing advertisers and other groups to focus on a specific subset of individuals based on their personality traits".

[11] OCEAN is a set of traits which psychologists use to measure and characterise personality types. The abbreviation stands for *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* and *Neuroticism*.

- Matz *et al.* (2017) proposed the analysis of Facebook user personality traits as means of mass persuasion to market more effectively to them. [10] In their work, Facebook users were required to fill in a questionnaire to determine their OCEAN personality traits. Once a user's traits were established, the researchers ran advertisements tailored to the users' personalities in a bid to correlate each individual's personality traits with their preferred advertisement style and format.

The work of these researchers is indicative of the extent to which detailed and sensitive information can be inferred from user models based on publicly available OSN data.

2.1.2.2.2 Expertise Inference

There have been multiple attempts in research to use OSN data to infer the expertise of users regarding particular topics.

- Z. Xu *et al.* (2011) tried to uncover the interests and expertise of Twitter users, proposing a novel topic modelling framework to do so. They constructed a dataset of 200 tweets, 200 retweets, 200 links[12], and 200 replies[13] and manually labelled each of them as either (i) topic-related or (ii) topic-unrelated. To uncover the underlying topics of interest in the Twitter data, they employed an extended variation of the Latent Dirichlet Allocation (**LDA**) algorithm which incorporates author information into the topic models[14]. [11]

  An interesting remark made by Xu *et al.* is that research into the temporal aspect of user profiling is an important area of future work: "…user interest[s] will change with time, in our model, we do not model the time factors explicitly". [12]

- C. Wagner *et al.* (2012) felt that users of OSNs can find "judging the topical expertise of other users in order to select trustful information sources about specific topics" to be difficult. [13] Focusing on the Twitter OSN in particular, their research used topic modelling to infer user expertise from tweets, finding that using tweets and retweets alone is "surprisingly useless" in inferring user expertise. Instead, they suggest that information related to a user's connections and group memberships provides more

[12] By this is meant that the corresponding 200 tweets included URLs.

[13] On Twitter, a *reply* is a response to another person's tweet. A user can reply by clicking or tapping the reply icon from a tweet.

[14] Topic modelling is a commonly used NLP and ML technique, which uses statistical modelling to discover topics that occur in a corpus of documents. It is commonly used for discovering hidden semantic structures in a body of text, as is the objective of techniques such as word embeddings which are described in *Section 2.3.3*.

accurate information regarding their expertise[15]. This suggests that though useful in user interest-based topic modelling exercises as found by Xu *et al.*, tweets alone may not prove useful in expertise modelling and will require some supplemental data enrichment processes.

- Y. Xu (2019) leveraged a combination of data from multiple OSNs to obtain enriched insights for inferring expertise, proposing a method of extracting data from several OSNs (Twitter, Quora[16] and LinkedIn) using (i) sentiment-weighted learning and (ii) topic relation-regularized learning. [14] These methods were used on multiple types of data from Twitter. As well as this, a multi-data and topic-relatedness combined learning model made use of Quora data, where users explicitly provide both their expertise domain and their Twitter account information.
  Xu's work is particularly interesting because of the way in which it considers datasets from several OSNs, and uses these to determine a user's expertise based on cross-referencing their public Twitter data with that of both LinkedIn and Quora.

2.1.2.2.3 Other Research using OSNs

Though not so numerous in number, some additional user profiling research works were encountered during the review of user profiling literature, outlined below.

- **User Preference Inference**
  In the research paper by R. Jiamthapthaksin *et al.* (2017), the researchers propose a means of modelling a user's Facebook preferences based on Facebook *page*[17] categories. [15] In the paper, the profile preferences were inferred with the use of ML models including Support Vector Machines (**SVMs**), Naïve Bayes and Artificial Neural Networks (**NNs**), based on the users' observable behaviour of liking and sharing pages and posts on Facebook.

[15] When considered, it is easy to understand why tweets and retweets may be considered 'useless' for the purposes of expertise modelling: Simply stated, tweets are noisy data points that contain information related to many facets of a user's life. Thus, they usually contain a greater proportion of personal data such as interests or relationships than data relating to expertise or profession.

[16] Quora is a website where users can ask, answer, edit and organise questions as a community. All of these activities are carried out on the basis of user opinion and expertise.

[17] On Facebook, a *page* is a Facebook profile which is owned and maintained by a business or interest group. As explained on their site, pages "make it easy for people to find out more about what you offer and get in touch". [81]

- **User Relationship Inference**

  In the research carried out by D. Tchuente *et al.* (2012) who proposed an egocentric network graph by using communities or friend lists the user has on Facebook. [16] The researchers propose that this technique can be adapted to perform in a variety of applications such as personalisation and recommender systems.

The sensitive nature of much of this research is impossible to ignore, though only explicitly addressed by a small number of the researchers. One of the co-authors of the controversial user preference research published by Matz *et al.,* Dr. M. Kosinski, wrote a letter explaining that the intended purpose of their research was to "illustrate the feasibility and effectiveness of the mass persuasion techniques deployed by companies... and exposed the risks inherent to behavioural targeting, the principal source of income for many tech giants". [17] Whether it is indeed true that the intended purpose was to inform policy makers of the potential risks of profiling methods, it is beyond doubt that there is definite potential for misuse of user profiling research, a topic which is the subject of the following section.

2.1.3 User Profiling and Data Privacy

In the domain of intelligent systems in recent years, there has been cause for great concern about user data privacy: Organisations both large and small have been the subject of scandals about the handling of their users' data privacy and the rights of their users regarding consent for use in user profiling applications. This section outlines the most significant data privacy scandal to-date in the domain of user profiling, which subsequently sparked action by some EU lawmakers to impose strict regulation on the use of consumer data.

2.1.3.1 Misuse of User Profiling

User profiling research is a hot topic not only in the technical realm in 2019, but also in public media. As already alluded to in previous sections, there is widespread concern about the motivations for conducting user profiling by organisations, and how the results are subsequently used.

The most recent and significant case of scandal in user profiling is best explained with reference to the paper published in 2017 by Matz *et al*. entitled "Psychological Targeting as an Effective Approach to Digital Mass Persuasion", mentioned previously in *Section 2.1.2.2.1*. These researchers stated that they would conduct their research "in a domain that is relatively

uncontroversial from an ethical point of view: consumer products", [10] and proceeded to outline the effectiveness of mass persuasion techniques.

Though not involved in the scandal, the naivety of the statements made by Matz *et al.* was demonstrated last year in 2018 when the now-infamous political consulting firm Cambridge Analytica was found to have capitalised on similar techniques to develop psychological profiles of millions of American Facebook users. These psychological profiles had been used to micro-target the users with more persuasive advertisements, a capability which was leveraged by US presidential candidates Donald Trump and Ted Cruz. The ultimate election of President Trump to the White House is a success widely attributed to the company as a result of implicitly influencing the population's voting decisions in his favour.[18]

The use of user profiling in this case demonstrates the potential for exploitation of such powerful insights to manipulate people without their knowledge, with such significant outcomes. Thus, concerns about ethical use of user profiling are well-founded, and it is clear that appropriate governance structures must be put in place to facilitate its correct and ethical use.

2.1.3.2 Governance of User Data Privacy

In all data-driven research, characteristics of data such as its origins, the legality of how it is obtained, and the purposes of its collection are all of significant importance to its ethical use. It is clear that individuals should not have their personal data to be gathered without their knowledge or consent, or allow for it to be used in the odious manner in which it was in the case of Cambridge Analytica's involvement in the 2016 US presidential election.

On May 25th 2018, the European Union implemented the General Data Protection Regulation (**GDPR**) in a bid to "protect all EU citizens from privacy and data breaches in today's data-driven world": A legislative document which is the first of its kind in the world. [18] [19] The GDPR sets out legally enforceable consumer rights to data protection with which organisations - termed "data controllers" and "data processors" depending on their involvement with user data – have responsibilities to implement. These rights encompass how consumer data of EU citizens is collected, stored, used and ultimately destroyed, whether being processed within the

[18] At the US congressional hearing for CEO of Facebook Mark Zuckerberg in April 2018 following the Cambridge Analytica scandal, US Senator Richard J. Durbin summarised succinctly the horror of the global community when he said "I think that may be what this is all about… your right to privacy. The limits of your right to privacy... And how much you give away in modern America". [92]

EU or abroad. All applicable organisations must adhere to these rules or face a fine of up to "4% of annual global turnover or €20 million (whichever is greater)" for violating the core principles of the regulations.

Though legislation cannot necessarily be equated to promoting ethical behaviours, the GDPR is heavily concerned with ethics. It speaks at length about the design of the regulation "to serve mankind" and to promote "...respect for private and family life, home and communications, the protection of personal data, freedom of thought, conscience and religion, freedom of expression and information, freedom to conduct a business, the right to an effective remedy and to a fair trial, and cultural, religious and linguistic diversity". Thus in respect of evaluating ethical concerns regarding the handling of user data, adherence to the GDPR provides a strong basis for ethical handling of user data.

2.1.3.3 Ethical User Profiling

Despite the obvious issues with improper use of user profiling, it is a domain which has clear potential for providing positive benefits to application users. Abdel-Hafez *et al* (2013) even go so far as to state that in the use of today's user-facing systems, "the need for personalisation increases dramatically to become a necessity". [20]

The importance of personalisation is only increasing as software systems continue to grow exponentially in size and complexity, becoming more and more difficult to navigate effectively. Personalisation is key to being able to provide great user experiences, delivering the most relevant and meaningful content to a user by tailoring it to their interests, attributes, moral values and lifestyle. This application of user profiling is with the intention of providing users with personalised interactions and content recommendations that help them maximise the value they get from software systems. As aptly phrased by A. Kobsa (2001), "since personalization has already been demonstrated to benefit both the users and the providers of personalized services… personalization is therefore going to stay". [21]

In a modern technology-engulfed society, it is difficult to find exemplars of user profiling and user modelling given that most OSNs give little consideration to user privacy. It is clear that the objectives of user profiling and user data privacy have little overlap and often contradict each other: Whilst user profiling is focused around harvesting user data, user data privacy is concerned with limiting access to user data.

- User profiling aims to take as much information as possible in order to build a descriptive model of a user. As stated by S. Schiffino *et al*. (2009) in their paper on intelligent user profiling, "a profile is a description of someone containing the most important or interesting facts about him or her… User profiling implies inferring unobservable information about users from observable information about them". [22]
- In stark contrast, user data privacy has the objective of honouring the fundamental human right to "a private life, to be autonomous, in control of information about yourself, to be let alone", by "ensuring the fair processing (collection, use, storage) of personal data". [23]

As discussed by F. Erlandsson *et al.*, (2012) there are many ways in which OSNs threaten the privacy of their users. [24] Such threats can range from OSN information leakage, to public information harvesting, to identity theft.

- OSN information leakage refers to where an OSN is continuously gathering its users' data, mining it to infer further information, and then selling the enriched data to third parties. This threat is typically the first privacy threat which OSN users worry about. [24] [25]
- Erlandsson *et al.* describe public information harvesting, which is where third parties which are not necessarily affiliated with the OSN can harvest user data due to it being publicly available.
- Identity theft is also a huge concern for OSN users, since OSNs provide an exceedingly easy means of committing identity theft. Malicious individuals can simply extract public photos, personal information, etc. from the profiles of legitimate users on an OSN, and create a false identity for themselves for any number of nefarious purposes.

There have been many papers published on the use of encryption to obscure data being transmitted over the internet, and based on this there have been examples of research that attempt to preserve the privacy of users in question through anonymisation and other means. [26] One such example of researchers trying to preserve user anonymity while still modelling preferences is the work completed by I. Dickson *et al.* (2003), who developed a user profiling system which claims to respect the privacy of the users by implementing control methods such as controlling (i) credentials and identifiers for user authentication, and (ii) which elements of the user profile are revealed and to whom i.e. access control. [27] Such work is crucial to the viable future of user profiling.

2.2 Temporal User Profiling

In the area of user profiling, temporal user profiling has been proposed as a means of capturing the inevitable shift in users data as they vary through time. The following sections give an overview into the area of temporal user profiling, illustrating the necessity of developing temporal models to better capture changes in user data through time, as well as providing some examples of interesting research papers and uses of temporal user profiling.

2.2.1 Overview

Temporality is an interesting aspect of user profiling, since people and their interests are not static: They change over time, often quite significantly. Arielle *et al.* (2013) demonstrated this through their research, which illustrated that an individual's musical interests vary through time, and tend to fluctuate and change around 'particular life changes' of the individual. [2] These findings affirm the inherent variability of user interests and preferences over time: The importance of capturing this aspect of users is an observation noted by Xu *et al.* (2011) in their research modelling the interests of Twitter users. [12]

Compared to static approaches, temporal user profiling has not been the subject of much research focus by the time of writing of this document. Most temporal modelling research has instead focused more indirectly on user profiling, such as that conducted by Holz *et. al* (2010) who investigated the change in meaning of language used in news articles. Rather than directly addressing the fact that individuals and societal groups' interests change over time, they simply note that "tracking the change of topics over time reveals interesting insights into a society's conceptualization of preferences and values". [28]

Despite this, there has been a recent marked increase in the volume of research into temporal user profiling, with a particular focus on analysis of OSN data. For such data, temporal analysis is highly appropriate since OSN users interact regularly with these platforms on an ongoing basis. Profiling of these users has the potential to produce enhanced user profiles, enabling predictions of elements such as the evolution of user behaviour, user interests, etc.

2.2.2 Research of Note

There are numerous examples of interesting temporal user profiling work which were encountered during this research. Although there are likely to be many more examples of such research conducted by major OSNs and related organisations, these have not yet been openly shared with the research community. Thus, all literature reviewed in this section is limited to what has been made openly available to the research community.

- In a paper by Zhang *et al.* (2012), the researchers propose a user profiling system that consumes mobile user data from the company *a Telecom*, modelling their users' browsing records from May to July of 2008 both dynamically and statically and then comparing the user models using different clustering algorithms. [29]
  The purpose of this research was to investigate the efficiency of different modelling and clustering techniques for *a Telecom* so the company could model users' data usage in a timely manner. Models proposed by the researchers are: (i) the batch model, which is a static implementation modelling the users non-temporally over the dataset; (ii) the evolving object model, which is an incremental version of the batch model where the final time series of the evolving object is the same as the batch model; and (iii) a dynamic data streaming model, which implements an ordered list that maps onto a map[19].
  The results of this research were rather interesting. Unsurprisingly[20], the dynamic data streaming model performed best when it came to temporally modelling user data usage; but surprisingly, the static batch model outperformed the evolving object model across all experiments in terms of clustering efficiency and quality.
- In their 2017 paper, Liang *et al.* proposed a dynamic user clustering topic (**DCT**) model which consumes Twitter data and generates a temporal model of the users as a vector, clustering the users whose models are similar. [30] The DCT model is a temporal extension of the Dirichlet Multinomial Mixture Model, which captures temporal aspects of short streams of text: In this case, tweets.
  The DCT model operates by (i) creating a model of a user at time t_i, (ii) creating a model of the user at time t_{i+1}, and then (iii) using the previous user model to construct the new user model. In this way, the DCT model attempts to better represent the change in the

[19] This operates such that when user data is updated, it is moved or added to the tail of the list, and when the list has reached a defined maximum size the head of the list is removed so that more temporally dynamic data may be added.
[20] This result is not very surprising, since by considering the data with respect to a period of time the users' data usage models would almost certainly be more complete and continuous in nature.

user model over time. In the paper, it is reported that the DCT model performs significantly better when compared to other topic modelling algorithms such as Language Model, Time-aware Microblog Search, LDA, Dirichlet Multinomial Mixture Model and Topic Tracking Model.

- Liang *et al.* (2018) developed a temporal user profiling system which consumed Twitter data. This system generated temporal embeddings[21] of both words and users, in a bid to model users' interests through time. [31] The researchers used an annotated Twitter data set, where the ground truths for evaluation were the users' interests for a given period of time. The embedding method implemented was a temporal extension of the skip-gram model from *Word2Vec*, which is described in *Section 2.3.3.2.2*.

  Practically speaking, their approach means that for any given time period a user could be compared to the words that occurred in the corpus. Therefore, for any given period of time for any user, the user's interests could be easily identified by comparing the user's vector with their surrounding word vectors using simple formulae like the cosine similarity rule. This approach of modelling users as vectors in the same vector space as their vocabulary is one of the novelties of this approach. According to the authors, theirs is the first instance of such an approach to temporal user profiling.

When considering the above research, it is evident that (i) temporal models tend to outperform their static counterparts, and that (ii) the methods discussed here are shown to outperform many other temporal modelling algorithms investigated by the relevant researchers. It is intuitive that temporal models should outperform static models of temporally varying entities such users, given that users themselves are not static and vary through time. The key finding here is that there is still an enormous amount of research that can be carried out into enhancing temporal user profiling.

2.3 Word Embeddings

Word embeddings are a vector space representation of words, used widely in Natural Language Processing (**NLP**). They are used to represent the semantic properties of the vocabulary of a corpus as vectors. An important NLP technique, they have been employed in domains as diverse as sentiment analysis, text similarity and machine translation. [36] This popularity can

[21] Temporal word embeddings are the subject of specific discussion in *Section 2.3.3.3*.

be largely attributed to their ability to encapsulate the context of a word in a document, the semantic and syntactic similarity of words as well as the relationship of a word to other words[22].

2.3.1 Overview

Vector space representations in language modelling came into prominence in the 1990s. Latent Semantic Analysis (**LSA**) was one of the first approaches to producing vector space models using a theory known as 'The Distributional Hypothesis[23]' (**DH**). [11]

In any application of a word embedding technique, there are two phases:

1. **Generation of Word Embedding Vectors**
   Word embedding vectors are generated differently depending on the technique: However, the result is always a set of words represented as vectors in some vector space.
2. **Analysis of the Word Embedding Vectors**
   After generating the word embedding vectors, further analysis must be performed in order to infer insights from them. This can include anything from comparison of the similarity of the vectors, to classification of vectors based on some defined criterion.

In the generation of word embeddings, it is important to understand the concept of a word's context. In simple terms, in a stream of text a word's *context* is the words that surround it. This context can sometimes be defined using a so-called *context window*, which is a context governed by a sliding window of size *n*, where *n* is the number of words adjacent to the current word.

[22] In fact, much of the interest around word embedding techniques comes from their applications in NLP tasks, where word embeddings are provided as input to a NN and can then be used to predict the next word(s) in a sentence.

[23] The Distributional Hypothesis is a theory which states that "*words which are similar in meaning occur in similar contexts*". [70] In more technical terms, the DH states that there exists a correlation between the distributional similarity of words and the meaning similarity of words: This correlation allows the use of the distributional similarity of the words to estimate the semantic similarity of words. This theory can be applied to NLP and other language-based tasks with good results. [71] As aptly phrased by linguist John Firth, "the complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously". [80]

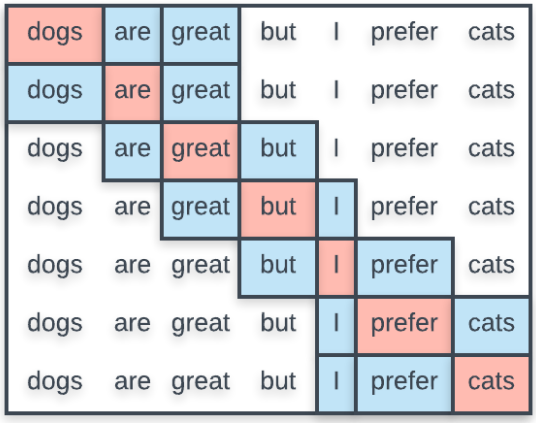

Figure 2-1: n-gram Generation.
This diagram illustrates how word n-grams can be generated using a context window. In this simple example, a context window of size 2 is applied to the sample sentence.

Figure 2.1 illustrates the generation of word n-grams using a context window. Each word is defined by considering the words in its context, i.e. the words either preceding or following the current word, or even a combination of both. The most basic technique to represent a word as a real-valued vector is the One-Hot Embedding (**OHE**) method[24]. In this approach, a single unit of text[25] is assigned a value based on its indexed position in a text document. An example of applying the OHE method to the text '*Look a cat*' is shown in *Figure 2.2*.

$$'Look' = [1,0,0]; 'A' = [0,1,0]; 'Cat' = [0,0,1]$$

Figure 2-2: One Hot Encoding Outputs.

The output of applying the OHE method to the text 'Look, a cat'. As shown, each word receives a vector value corresponding to its index in the stream of text

An entire stream of text can be represented as a *co-occurrence matrix*[26], where each of the word vectors forms a row of the matrix, as illustrated in *Figure 2.3* for the text 'Look a cat'.

[24] In literature, this technique is often referred to by the equivalent name 'One-Hot Encoding'.
[25] This unit of text is generally one of a character, a word, or a sentence.
[26] Sahlgren describes the co-occurrence matrix F as follows: "…each row $\boldsymbol{F_w}$ represents a unique word $\boldsymbol{w}$ and each column $\boldsymbol{F_c}$ represents a context $\boldsymbol{c}$... The cells $\boldsymbol{F_{wc}}$ of the co-occurrence matrix record the frequency of co-occurrence of word $\boldsymbol{w}$ and document or word $\boldsymbol{c}$. As an example, if we use document-based co-occurrences, and observe a given word three times in a given document in the data, we enter 3 in the corresponding cell in the co-occurrence matrix." [39]

Look =	1	0	0
A =	0	1	0
Cat =	0	0	1

Figure 2-3: One Hot Embedding Matrix
When the OHE vectors are produced, each vector can then be stacked, producing a matrix representation of the documents that were processed.

For a very large corpus, OHEs for words can theoretically produce vectors (and hence matrices) of extraordinarily large dimensions, with a theoretical maximum of approximately 13 million[27] dimensions: Far too large to be computationally economical when it comes to analysis. Given this limitation, state-of-the-art word embedding techniques reduce this dimensionality to make vector operations more computationally manageable whilst retaining the information contained therein.

2.3.2 Popular Techniques

Since the 1990s, there has been much research in the area of word embeddings. Some of the most popular state-of-the-art techniques used today are Word2Vec, Latent Semantic Analysis, Random Indexing (**RI**), Global Vectors for Word Representation (**GloVe**), and FastText.

2.3.2.1 Latent Semantic Analysis

LSA is a commonly used word embedding technique devised by Deerwester *et al.* (1990), [37] also commonly referred to as Latent Semantic Indexing. By using a technique called truncated Singular Value Decomposition (**SVD**), LSA can be used to reveal underlying semantic structures in data that may be "partially obscured by the randomness of word choice with respect to retrieval". [38]

As noted by Sahlgren, LSA "has become a household name in information access research, and deservedly so; LSA has proven its mettle in numerous applications, and has more or less spawned an entire research field since its introduction". [39] There are however significant limitations to LSA: In particular, SVD dimensionality reduction does not scale efficiently, and

[27] Merriam Webster tallied some 470,000 entries in the Webster's Third New International Dictionary, Unabridged, together with its 1993 Addenda Section. This tally is likely not to have taken tenses into account, which explains the disparity between the theoretical maximum of 13 million and the reported 470,000 figure. [72]

thus is a time and compute-intensive process. As well as this, the performance of the technique has been found to depend heavily on the structure of the corpus. [40] Despite this, LSA remains an important and widely-used technique with many useful applications in signal processing, statistics and NLP.

2.3.2.2 Word2Vec

Word2Vec was the first globally popular NN-based word embedding model, introduced by Mikolov *et al.* (2013). [41] It is a log-linear classifier, based on two simple models: Continuous Bag-Of-Words (**CBoW**) and Skip-gram.

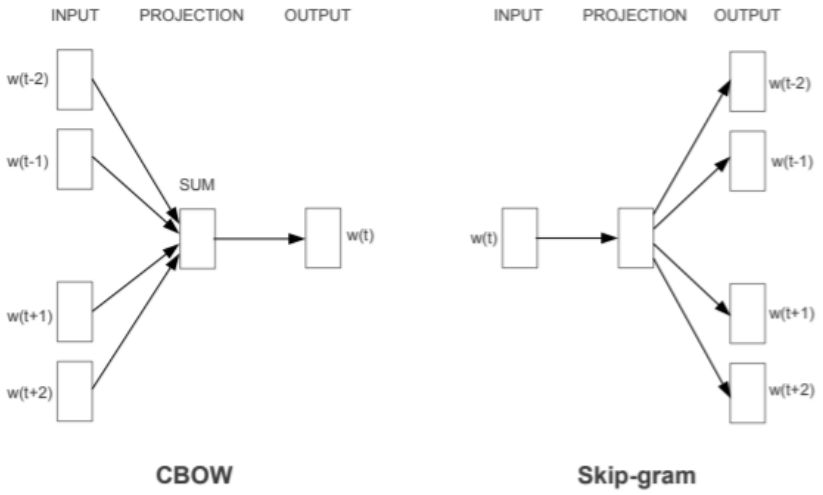

***Figure 2-4:* Word2Vecs CBoW and Skip Gram models**
A diagram taken from the original Word2Vec paper published by Mikolov et al., showing the NN architecture of the Word2Vec model. [41]

- The CBoW model is based on a feed-forward NN language model, whose objective is to predict the current word using its surrounding context by minimising a defined loss function.
- The Skip-gram model operates similarly to the CBoW model, but with an opposite objective: It "tries to maximize classification of a word based on another word in the same sentence", predicting the surrounding context words of a given target word. [41]

As shown in *Figure 2.4*, the Word2Vec NN architecture consists of input, hidden/"projection", and output layers.

- In the case of the CBoW model, the input layer corresponds to the context of the target word, taking the context of the word to be the combination of OHE vector

representations of the surrounding words. The output layer has the same dimensionality as the input layer, and contains the OHE vector representation of the target word.

- In the case of the Skip-gram model, the opposite is true: The input layer corresponds to the target word, and the output layer corresponds to the context of the target word.

When compared to techniques such as LSA, Word2Vec consumes much less memory since it doesn't involve generation of a highly-dimensional co-occurrence matrix as its first step. As discussed by Mikolov *et al.*, Word2Vec scales well to large datasets: However, in the case of the CBoW model the accuracy is not as high for frequently occurring words on a smaller corpus. [41] Though often criticised for the fact that as a predictive technique, it requires a large input dataset to perform very well, Word2Vec outperforms many other word embedding models[28] in a variety of NLP tasks.

2.3.2.3 Random Indexing

RI is a word embedding method proposed by Magnus Sahlgren (2004). [39] It operates by generating vector representations of words in an *N*-dimensional vector space and constructing a co-occurrence matrix, similarly to LSA and several other techniques. It is used to generate a "good"[29] vector representation of words occurring in a given corpus.

In contrast to methods such as LSA, RI is an incremental process that takes inspiration from Pentti Kanerva's work on sparse distributed representations, and does not require a separate dimensionality reduction step for the co-occurrence matrix. [43] [44] Instead, so-called *context vectors* are built incrementally by accumulating words in defined context windows.

[28] In research by Schnabel *et al.*, the CBoW Word2Vec model outperformed the C&W, Hellinger PCA, GloVe, TSCCA and Sparse Random Projection word embedding models. [42]

[29] A vector representation would be considered good provided it has sufficient context to build a general, well-contextualised vector representation.

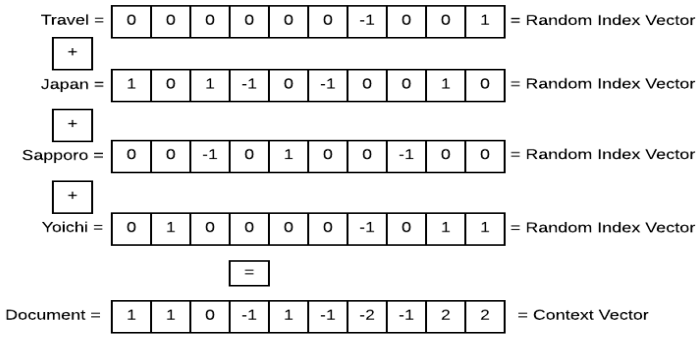

Figure 2-5: Random Indexing Context Vector Generation
This image provides an illustration of how, given a series of words used to define another word or an entity, RI is performed. Initially, each word in the corpus is assigned a random index vector. Every time a given word occurs within the context of the current word, its index vector is summed with the other words in the context to generate a context vector. The example above could be used to represent Hokkaido, the northern most island of Japan, where Sapporo is the capital city frequented by travellers due to the abundant wildlife and the Yoichi Whisky Distillery

The result of the RI operation is an approximation of the co-occurrence matrix *F'*: However, without requiring an explicit dimensionality reduction step it can achieve the same dimensionality reduction effects as the SVD algorithm in the LSA method[30]. Overall, RI has been found to have multiple advantages over other word embedding techniques: it is (i) usually less compute expensive, and (ii) and doesn't require access to the whole term-document frequency matrix as it is incremental in nature. It has also been found to outperform both LSA and Word2Vec by some researchers. [45]

2.3.2.4 Global Vectors for Word Representation

GloVe is a unsupervised word embedding technique, first proposed by J. Pennington *et al.* (2014). [46] It operates by learning real-valued vector representations for words using a weighted bi-linear regression model, making use of global co-occurrence statistics derived from a training corpus. The algorithm involves the construction of a co-occurrence matrix, before applying a least squares minimisation to it as part of an explicit dimensionality reduction process[31].

When applied to a stream of text, the resulting vector space demonstrates some interesting properties:

[30] Rather than using a complex algorithm such as SVD to perform dimensionality reduction, RI's implicit dimensionality reduction makes use of simple vector addition operations.
[31] Prior to dimensionality reduction, a weighting function is applied to the cost function of the model to reduce the impact of high frequency co-occurrences that occur a greater distance away from the current term *j*.

- Linear substructures can be observed, i.e. the difference between similar terms such as *man* and *woman* are found to be approximately be equal to the difference between terms such as *King* and *Queen*.
- A number of "nearest terms" can be observed for any term, i.e. if one was to use GloVe in a search for the top k-nearest terms to *frog*, the terms returned might be *frogs*, *toad*, *litoria*[32], etc.

Given this, it is no wonder that GloVe tends to perform particularly strongly in analogy and similarity tasks: A strength due to its approach of learning similar vector representations for terms which appear in similar contexts[33]. [47]

Similarly to both LSA and RI, GloVe learns its vector representations based on first generating the entire co-occurrence matrix: An operation which consumes a lot of memory compared to techniques such as Word2Vec, and which is much more compute-intensive than techniques such as RI. However, GloVe and Word2Vec undergo a similar process for dimensionality reduction. [48] Ultimately, Word2Vec has been found to be marginally more accurate and approximately half as performant as GloVe, but consumes almost 15 times less memory than GloVe. [48]

2.3.2.5 FastText

FastText is a NN-based word embedding model, built upon the work done by Mikolov *et al.* in *Word2Vec* as an extension to the CBoW model. It was developed by Joulin *et al.* as part of Facebook's AI Research (**FAIR**) lab. [50]

FastText makes use of n-grams to produce a series of vector representations for each token-word. This concept can be seen in *Figure 2.6*. It is most notable because it is a simple linear classifier, which has been found to achieve the same accuracy of much more complex deep learning algorithms, but orders of magnitude faster. [51] Because of its use of n-grams, FastText is able to take into consideration the morphology of a word[34]. The result is the

[32] *Litoria Splendida*, usually shortened to *litoria*, is a species of frog, more specifically the Australian tree frog native to Western Australia.

[33] Unlike some other methods, GloVe computes the co-occurrence frequencies of terms within the context of the entire corpus rather than context windows.

[34] Morphological Word Embeddings (**MWE**s) are word embeddings which take the morphology of words into consideration when trying to generate a good generalised vector representation of the words in a given corpus. For example, the verb *to sit* can take the forms *sit*, *sat*, *sitting*, etc. In this way, FastText is able to conflate morphologically similar words like "*watches*" and "*watch*".

generation of multiple representations for a single word by treating each given word as a sum of character n-gram representations.

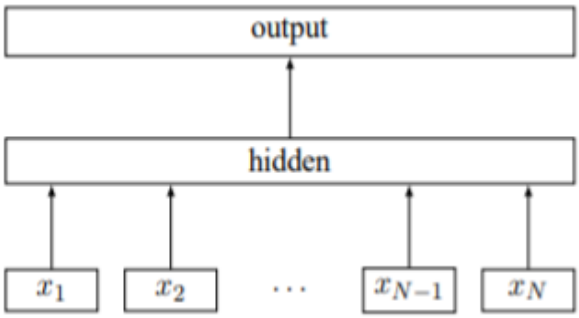

Figure 2-6: **FastText Model Architecture**
A diagrammatic illustration of how the k n-gram vectors, labelled x_n to x_{n-1} are provided as input to a NN for the purposes of the dimensionality reduction step involved in the CBoW implementation in FastText. This diagram is taken from the FastText research paper. [50]

Similarly to Word2Vec, word vectors in FastText are generated typically using OHE by breaking the text document into n-grams. The result of this is a set of vectors with high dimensionality, where each n-gram receives a vector value corresponding to its index in the stream of text. An explicit dimensionality reduction step is then performed using a NN[35] as in Word2Vec. Once the FastText NN model has been trained, it will have learned all of the n-gram representations that existed in the training corpus. This means that infrequently occurring words can now be generated with good representations using the n-gram vectors, since it is highly likely that some of the n-grams present in the infrequently occurring words will also appear in other words that were present in the training corpus.

As reported by Bojanowski, the results for the FastText model show that the performance of the model is very good, outperforming multiple state-of-the-art approaches whilst approximately matching that of others. [52] [53] [54] It is also noted that the performance of the model increases as the number of n-grams increases: When both bi-grams and tri-grams were used, the accuracy reached 97.1% on a given dataset.

2.3.2.6 Conclusions

It is clear from analysis of the five word embedding techniques described in this section that there are a number of characteristics that must be considered before selecting a word embedding for a given application.

[35] In the hidden layer of the NN, the model reduces the dimension of the vectors from *m* to *d* such that $d << m$. Thus, the dimensionality of the resulting matrix is greatly reduced.

- Performance is a key consideration, since it is always desirable to have the highest levels of accuracy and precision in the generation of word vectors for a given task.
- Scalability is another important factor: Depending on the application for which word embeddings will be used, this property may restrict the practical usability of certain techniques.

Table 2.1 compares the five word embeddings discussed in this section based on three characteristics: (i) whether or not the technique involves initial generation of a co-occurrence matrix, (ii) whether or not the technique includes an explicit dimensionality reduction step, and finally (iii) whether the technique is predictive which generally results in greater performance for a larger dataset.

	Explicit Dimensionality Reduction Step[36]	**Initial Generation of Full Co-Occurrence Matrix**[37]	**Predictive (ML-based) Technique**
LSA	Yes	Yes	No
Word2Vec	Yes	No	Yes
Random Indexing	No	Yes	No
GloVe	Yes	Yes	Yes
FastText	Yes	No	Yes

Table 2-1: Comparison of Static Embedding Techniques
This table provides a comparison of some of the primary characteristics of the LSA, Word2Vec, RI, GloVe and FastText word embedding techniques

2.3.3 Temporal Techniques

Temporality with respect to word embeddings deals with the way in which word semantics change over time. There is increased interest in this variety of word embedding, stemming from the fact that there is now an abundance of time-variant data available from major websites and application platforms, as well as an increasing understanding that word meaning does not remain static. *Figure 2.7* illustrates how the meaning of words can change drastically over even short time periods.

[36] The implication of a 'Yes' in this case is the indication of a less scalable technique which is not incremental in its generation of embedding vectors.
[37] The implication of a 'Yes' in this case is the indication of a less memory-efficient technique, since the generation of a full co-occurrence matrix of potentially very large dimensionality implies large consumption of memory.

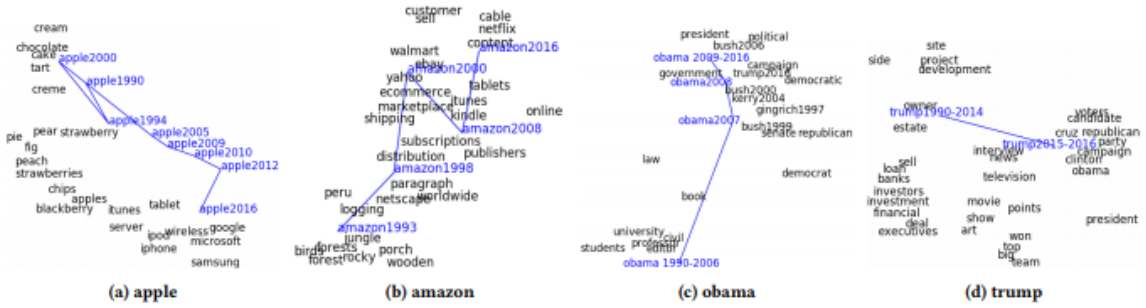

Figure 1: Trajectories of brand names and people through time: apple, amazon, obama, and trump.

Figure 2-7: Temporal Word Analysis

An illustration of how word meanings change over time. For example, the word 'apple' in 1994 had a similar semantic meaning to 'pear' and 'strawberry', which are all fruits. By 2016, the same word 'apple' is seen to be most closely associated with 'google', 'microsoft' and 'tablet', demonstrating how this word is now more semantically associated with technology companies. Similar evolutions in word meaning are shown for 'amazon', 'obama' and 'trump'. Original source: Yao et al. [54]

Revisiting the domain of user profiling research, it is clear that understanding the temporal aspect of words and their semantics is a problem of major interest. Word embedding techniques are a strong candidate for being able to solve this problem. There have been several methods proposed which extend temporally previously static embedding methods.

2.3.3.1 Temporal Random Indexing

Jurgens and Keith (2009) proposed an approach to extending the static RI word embedding model such that it would capture temporal information. [56] The result was Temporal Random Indexing (**TRI**), which generates word embeddings as a function of time, enabling analysis and investigation into the evolution of word meanings over time.

To capture the temporal aspect of word embeddings, the word vector generation process of RI is altered. This requires that the context first be annotated with timestamps, to allow for capture the meaning of words at a given time for a given time period[38]. For each period, a separate word space is produced. For this, a range of *time bins* must be defined, e.g. a time bin could be years, days, weeks, etc.

TRI has been applied to temporal word embedding problems with promising results. Two of the more significant cases in this regard are discussed below.

[38] Such a time period could be a day, week, month, year, etc. This is not the same as a time *bin*, which is an instance of a time period: For instance, given a time period of 1 year, an instance of a time bin for this time period would be the year 2018.

- In the original TRI paper by Jurgens and Stevens, [56] the authors used TRI to automatically detect novel and interesting topics as they are published on the internet in a time series of months. They used TRI as a method for effectively detecting new events in blog posts by evaluating the semantic shift in a set of key terms over time.

  To understand this, consider the term *Lebanon*. As stated in the paper, the terms found to be most closely associated to *Lebanon* were names of other countries. However, a war broke out in Lebanon in 2006, an event which was detected by TRI as a shift in meaning of the word *Lebanon*. The result of this was that the term *Lebanon* became more closely associated with terms such as *Hezbollah, soldiers* and *rockets*: A clear and dramatic shift in the class of words similar to *Lebanon* pre-2006. This example clearly illustrates how TRI can be used to effectively capture the temporal evolution of word semantics.

- Basile *et al.* (2016) proposed the use of TRI in order to analyse word meaning variations in news articles, enabling the identification of linguistic variations which emerge in specific time intervals. [57] Such identified variations could then be related to particular events reported in the news.

  Using TRI, the researchers were able to measure the change in a term's vector representation over time, correlating the change in meaning and use to events that occurred in news articles. One of the examples used to illustrate this in their paper was when Volkswagen, the automotive manufacturer, was hit by a diesel efficiency cheating scandal in 2016: It was shown that during the time of the scandal, the semantic similarity between the terms *scandal* and *Volkswagen* increased. Additionally, in an IR test scenario comparing TRI to Vector Space Model (**VSM**), they proved that TRI was better able to return more relevant documents relating Volkswagen to the emissions scandal at the time, using only the query "scandal Volkswagen"[39,40].

These works demonstrate the excellent performance of TRI in temporal word embedding tasks: Indeed, Basile *et al.* state that they chose to temporally extend RI since (i) "the method is incremental and requires few computational resources while still retaining good performance",

[39] Interestingly, the articles retrieved in this task when using the TRI approach contained no mention of the term *scandal*.

[40] This can be explained based on the fact that the semantic search used with TRI works based on ranking the results based on their proximity to the submitted query.

and (ii) "…the methodology for building the space can be easily expanded to integrate temporal information". The ease of implementation and quality of performance of the resulting TRI technique is an encouraging sign for the potential of applying this approach to temporal user profiling.

2.3.3.2 Temporal Word2Vec

Yao et al. (2018) recognised the increasing importance of temporal word embedding models, and proposed "a dynamic statistical model to learn time-aware word vector representation". [54] Their research highlights the shortcoming in current approaches, where "recent advances such as word2vec and GloVe… usually do not consider temporal factors, and assume that the word is static across time". [54]

Their research proposes the use of dynamic Word2Vec and compares there dynamic Word2Vec implementation to three other static Word2Vec implementations; namely

1. statis-Word2vec, which is the standard Word2Vec implementation,
2. Transformed-word2Vec,
3. Aligned-Word2Vec. From the results obtained in their research, Yao *et al.* illustrated that their dynamic Word2Vec implementation outperforms its static counter parts.

Liang *et. al,* previously referenced in *Section 2.2.2,* also proposed a temporal extension of the Balmer-Mandt's Skip Gram filtering technique in their Dynamic User and Word Embedding (**DUWE**) model. In their paper, the researchers compare their model to several techniques. Of note to this section are two approaches used to extend Word2Vec temporally:

1. The Dynamic Independent Skip-Gram model, which splits the dataset into separate time bins, independently initialises word representations and obtains the word embeddings for each time bin using Word2Vec, [41] [58] and
2. The Dynamic Pre-Initialised Skip Gram model, for which the approach is the same as in the Dynamic Independent Skip-Gram model; however, the word vectors for time bin *t+1* are initialised with the vector values from the previous time bin *t*. [58]

2.3.3.3 Dynamic Predictive Language Model

The dynamic predictive language model learns the dynamics of personal interests based on a probabilistic language model which is used to model how a person's expertise either changes

or remains static there expertise. The work completed by Fang *et al.* illustrate the effectiveness of this model in dynamic user personal expertise. [59] The model was applied to data obtained from ArnetMiner[41] and from the data, the researchers were able determine a user's given expertise as they vary through time periods of years.

In their research, there were three primary factors considered relevant to model:

- The personality type of the given expert when exploring new domains of expertise,
- The similarity between the new domain and the expert's current domains and,
- The popularity of the new expertise domain. Their research found that their Dynamic Predictive Language Model outperformed all baselines models considered.

2.3.3.4 Dynamic Clustering Topic Model

In the work completed by Liang *et al*. in inferring dynamic user interests for user clustering, they proposed a dynamic extension to the Multinomial Dirichlet Mixture Model for the purposes of clustering users based on shared interests inferred from their vocabulary. [30] It was found that his technique could accurately capture a user's time-varying topic distributions in short streams of text such as data from micro-blogging websites like Twitter.

2.4 Summary

As evidenced by the state-of-the-art in techniques, approaches and research explored in this chapter, it is obvious that the domain of user profiling is nascent in its development, and open to improved solutions.

Many of the challenges that face researchers in fields such as IR, where modelling users interests for the purpose of content personalisation are highly desirable, exist due to the temporal nature of the world and how human users change over time. In the field of user profiling, modelling the temporal aspects of users and their words simultaneously is little studied but is clearly gaining traction as can be observed from the research literature. It is the opinion of the author that all of these challenges present an opportunity for improvement to the state-of-the-art in temporal user profiling.

[41] ArnetMiner is a free, online service which is used to index, search and mine large quantities of scientific data.

Chapter 3 Design

This chapter describes the formulation of the research question based on the literature review conducted in *Chapter 2*, with consideration given to design decisions and the challenges which were encountered at the design stage of the research.

Section 3.1 describes the formulation of the research questions, identifying gaps in current research literature and from this, developing a set of research objectives.

Section 3.2 details the initial design considerations and decisions which were made to address both the functional and non-functional requirements of the user profiling research solution, as well as the design of the evaluation stage.

Section 3.3 discusses the preliminary challenges which were uncovered and addressed, with particular emphasis placed on the dataset used.

Finally, *Section 3.4* provides a brief summary of the chapter.

3.1 Problem Formulation

It is clear from the discussion of the state-of-the-art in *Chapter 2* that numerous challenges exist in the field of user profiling. From the research literature reviewed, one thing is abundantly clear: There has been scant research conducted into the temporal aspects of how users and their interests vary through time, particularly with respect to how they relate to the vocabulary of OSN users. It is this observation which motivates the formulation of the research question upon which this work is based.

3.1.1 Opportunities Identified

Although there are many approaches to user profiling outlined in *Section 2.1* which have contributed excellent work toward the advancement of the user profiling domain, there exist many opportunities for improvement. Some significant opportunities for improvement identified during this research are discussed below.

3.1.1.1 Temporal User Profiling

As noted in *Chapter 2*, research conducted into temporal user profiling hasn't seen the same degree of interest as static user profiling, to-date. It is the opinion of the author that this is a shortcoming of current approaches, since the enhanced potential insights achievable from such analysis are almost entirely neglected: a fact which has been recognised by multiple researchers. This presents a significant opportunity to make a contribution to user profiling research.

3.1.1.2 Use of Embeddings in User Profiling

There has been very little research identified during the state of the art review, presented in Chapter 2 which explores the use of embeddings to model and represent users either statically or dynamically. As discussed in *Section 2.2.2*, the research conducted by Liang *et al.* shows promising results for the use of embedding methods in user profiling: Their work (i) illustrates that temporal extensions to static embedding methods tend to perform better than static, non-temporal embedding methods such as the plain Word2Vec skip-gram model, whilst also (ii) giving rise to an interesting and novel way in which users can be modelled and profiled.

It is clear that word embedding techniques have the potential to be a powerful tool in the arsenal of user profiling. It is also clear that there are many challenges that exist: In particular, the varying scalability of word embedding techniques, and the gap in the ability of existing user profiling techniques to account for the temporality of user data. It is the opinion of the author that there is a significant opportunity for word embedding techniques to be adapted to better fulfil the needs of user profiling applications.

3.1.2 Research Question

Based on the opportunities for improvement outlined in *Section 3.1.1*, a research question was devised, as follows:

To what extent can a user's vocabulary on an online social network be used to infer their interests as they vary through time, using word embedding techniques?

This is clearly a broad question: Thus, a number of specific objectives were also defined which when achieved, would contribute to answering the proposed question. The objectives defined are as follows:

1. To select an appropriate word embedding technique to address the research question. Such a word embedding technique should demonstrate high potential for temporal analysis of OSN user data.
2. To devise an effective approach to representing users by the word embeddings generated from their OSN data**.**
3. To define an appropriate evaluation approach which would allow for accurate comparison of the resulting user embedding technique against other similar methods.

The formulation of the research question and objectives provided a concrete direction in which to take the research. Subsequent design decisions were made to aid in achieving the objectives, and are described in the following section.

3.2 Design Considerations

This section describes many of the functional and non-functional design considerations and decisions made relating to the implementation of the final research solution.

3.2.1 Dataset Considerations

This section describes the initial design decisions regarding the choice of dataset, word embedding technique used and data processing required.

3.2.1.1 Choice of OSN

As discussed in *Section 2.1.2*, there has been a great deal of research conducted in the profiling of users of various OSNs. However, Twitter is the most commonly used OSN for such research, which has long been the case. Twitter's popularity and suitability for user profiling research is largely attributable to three factors, [60] as follows:

1. **Simplistic Data**

   When compared to data from other OSNs, data on Twitter is comparatively simplistic in content and structure. The *tweet*, which is the base unit of information on the platform, has straight-forward attributes[42] which translate to a simple data object with well-defined fields.

[42] A sample selection of simple tweet attributes include i) user Id, ii) time stamp, iii) likes, iv) retweets, v) replies, vi) embedded content, all of which are contained in a tweet.

2. **Open Data Access**

   The fact that much of Twitter's data is made publicly available through their Application Programming Interfaces (**API**s) makes it easy to access their data. [62] Such data is shared based on having obtained consent from the relevant users[43].

3. **Micro-Blogging: Expression of Opinion**

   A final factor is the micro-blogging nature of how users behave on the Twitter platform. Most tweets are an expression of a feeling or opinion rather than being purely used for content-sharing, [62] adding to their value in user profiling and related applications.

OSNs such as Facebook, Instagram, Reddit and LinkedIn are less well-equipped to provide data for user profiling research in general: The comparative heterogeneity and complexity of data artefacts, as well as huge inconsistency of user behaviours and data access policies, make it overall more difficult to conduct research using their data. Thus, Twitter was selected as the most appropriate OSN for the research.

3.2.1.2 Choice of Dataset

Given that Twitter had been selected as the most appropriate OSN for use in this research, an appropriate Twitter dataset needed to be either generated or obtained. It was decided that given the limited time within which to complete the research and that a relatively large dataset is required, obtaining a Twitter dataset which had already been published for use in research would be the most feasible option.

Upon a review of the Twitter datasets used in some of the user profiling research referenced in *Chapter 2*, it was identified that the Twitter dataset collated by Liang *et al.* in their development of the UCT model [30] was well-suited to use in this research:

- The dataset was collected by University College London's Big Data Institute, and subsequently used by Liang *et al*. in multiple temporal user profiling research projects, [30] [31] one of which also involved the use of temporal word embeddings.

[43] This is a potential concern of using the any OSN for research, since there are inherent privacy risks in making user data publicly available through APIs. According to their Privacy Policy, Twitter obtains consent from its users before making such information publicly accessible. [65]

- The dataset is openly available[44] and suitably large, capturing "1,375 active users that were randomly sampled from Twitter", with approximately 3.78 million tweets.
- The dataset is suitable for temporal user profiling research as-is, due to "each tweet (in the dataset) having its own timestamp".
- Based on initial inspection, ground-truth data which would be required for evaluation of a system using this dataset had also been made openly available by the researchers.

Given these characteristics, it was decided that the UCL dataset would be used in this research.

3.2.1.3 Choice of Word Embedding Technique

In order to address the formulated research question, a promising word embedding technique needed to be selected. Upon consideration, the temporal RI word embedding technique was found to fulfil a number of key requirements:

1. RI is a simple, scalable word embedding technique, which is much less time- and computationally-intensive than many of its counterparts that use a co-occurrence matrix due to its implicit dimensionality reduction[45] and its incremental approach to constructing the co-occurrence matrix[46].
2. There is pre-existing research into the use of RI for temporal analysis, including the development of TRI. This research demonstrates the ability of TRI to accurately capture temporal variations in word usage. However, no research has been identified which applies this temporal word embedding technique to the domain of user profiling, an open opportunity which has potential benefits for the research community.

[44] The researchers in question made the dataset publicly accessible through Shangsong Liangs BitBucket account. [93](Accessed 6th April 2019)

[45] Given the use of a fixed dimensionality *d*, which is significantly smaller than the number of all possible contexts *n* present in the corpus, there is vastly less computation required to produce a word vector using RI when compared to explicit dimensionality reduction processes such as SVD in LSA word embedding, the process for which is illustrated in Figure 3-1.

[46] The fact that RI is also incremental means that results can be produced before the entire corpus has been analysed, providing huge additional time efficiency benefits. This also means that new data can be introduced whilst RI word embeddings are being generated without affecting the generation of word embeddings based on the prior data.

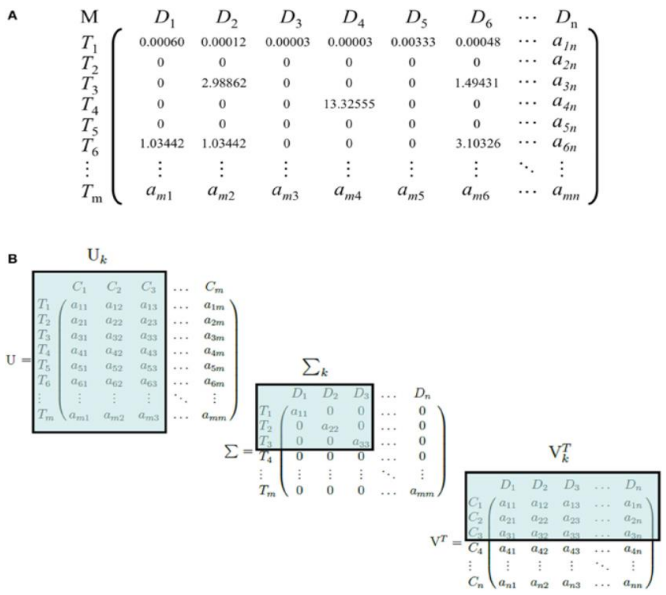

Figure 3-1: LSI Dimensionality Reduction

A diagram illustrating one of the weaknesses of the LSA word embedding technique when compared to RI: Explicit dimensionality reduction. As a batch process requiring access to the entire corpus prior to generating the vectors, it is much less scalable than the implicit dimensionality reduction process in RI. [63] In the above diagram, A corresponds to the co-occurrence matrix; generated similarly to RI, however LSA implements an explicit dimensionality reduction process; SVD, as illustrated by B in the above diagram.

The fact that the dimensionality reduction process is implicit and incremental in RI is a major strength of RI-based methods: From the perspective of web-scale applications, it has the potential to provide scalability and time efficiency that is difficult to find in many other co-occurrence word embedding techniques. Given these considerations, TRI was selected as the best temporal word embedding technique for the purposes of addressing the research question.

3.2.1.4 Data Preparation Considerations

Upon inspection of the selected dataset, it was clear that the data would need to be pre-processed in order to make it suitable to use in word embeddings. Several text normalisation techniques were selected for this purpose, as follows:

1. **Lemmatisation**

   Lemmatisation is a process whereby text is converted from words of a sentence to its dictionary form. Given the same example used to explain stemming above, the lemma for each of *amusement*, *amusing*, *amused* would be *amuse*.

2. **Stop Word Removal**

   Stop word removal is a process whereby any words that are a member of a so-called *stop word list* are removed from the corpus of text. This is often applied to words that occur extremely frequently in a corpus, such as *and*, *the*, etc.

3. **Text Normalisation**

   Text normalisation is a series of processes concerned with transforming text into one single canonical form. For example, one such process would be to remove so-called *diacritical marks* that are present in languages such as the character ‘ñ’ in Spanish.

3.2.2 Functional Considerations

This section describes the functional design decisions made regarding the data storage mechanism, the implementation of the temporal word embedding technique, and the development of the evaluation environment.

3.2.2.1 Data Storage Mechanism

For the purposes of this research, it was identified that a JSON-based In Memory Database (**IMDB**) would be the most suitable option for storage of the data.

- Since the UCL dataset discussed in *Section 3.2.1.2* is in JSON format, use of a JSON-native database would reduce complexity in the querying and processing of the data by the user profiling system. As well as this, JSON databases are generally schema-less, eliminating the need to construct and normalise a relational database in order to efficiently execute queries.
- IMDBs by their nature are capable of providing data faster than other storage options, reducing the amount of time required to read in the data from memory compared to other storage options.
- It was realised early on that the data would need to be pre-processed in order to be in a form suitable for temporal user profiling[47]. Thus, the use of a database for storage as opposed to the use of the as-is data in a file would also eliminate the need to re-process the data each time the system is launched and storing the data in an IMDB would allow for time improvements over reading and writing to text further illustrating the design of implementing such a storage method into the system.

Thus, a MongoDB IMDB was selected for use in the user profiling system[48].

[47] This pre-processing is discussed later in *Section 3.3*.

[48] MongoDB is an industry-standard NoSQL database, which stores data in JSON format. It is open-source, highly scalable and schemaless, making it excellent for use in web-scale applications. [94]

3.2.2.2 Implementation of Temporal Random Indexing

One of the primary decisions to be made regarding functional design of the temporal user profiling system was how the TRI word embedding technique would be implemented.

It was decided that given the limited time within which to complete the research, permission would be obtained from Basile *et al*. to build upon their implementation of a TRI system in the Java programming language, previously discussed in Section 2.3.3.1[49]. These researchers had open-sourced their implementation of TRI, meaning that it could be extended with relative ease compared to what would be required in implementing such a system in its entirety[50].

Once this decision had been made and the appropriate permission obtained, there were a number of additional factors to be considered with regards to extending their implementation.

- The system implemented by Basile *et al.* was developed with the objective of conducting temporal analysis of news events based on reading from files in a non-standard format. In contrast, the analysis being performed in this research would read JSON data stored in a MongoDB IMDB to conduct temporal analysis of OSN users.
- In Basile *et al.*'s implementation of TRI, the system was designed only to process data separated into time bins of years. In the system being proposed as part of this research, it was identified that other time bins would need to be analysed in order to perform an accurate evaluation against the state-of-the-art in approaches.

Thus, it was decided that (i) the code written by Basile *et al.* for reading the input dataset would need to be replaced with calls to a MongoDB IMDB instance, (ii) the TRI analysis in the code written by Basile *et al.* would need to be extended to handle input data separated into time bins other than just years, and (iii) the TRI implementation would also need to be extended to handle the generation of user embeddings.

3.2.2.3 Implementation of a Temporal User Profiling System using TRI

Another major consideration in the design of the temporal user profiling system was the way in which users could be profiled based on the TRI word embeddings that would be produced. In the implementation of this research, it was decided that the most appropriate approach to

[49] The required permission was obtained from A. Caputo, co-author of the 2016 paper "Temporal Random Indexing: a Tool for Analysing Word Meaning Variations in News". [57]
[50] In addition, there was no library support found in any implementation language for the RI model.

generating a user vector was to model it based on the language explicitly used by a user. To this end, it was devised that to generate a user vector, it would be based upon calculating the centroid of the tweets written by a given user; allowing each tweet to modelled based on the language contained in the tweet. From this, the next step is to generate a final user vector, which will be obtained by calculating the centroid of each users tweets to obtain a user vector. This would result in model of every user based on the words used in their tweets.

Practically speaking, to achieve this based on the TRI implementation developed by Basile *et al.* would require additional development work, as follows:

1. The co-occurrence matrices generated as part of the RI process would need to be updated to handle tweet-word co-occurrences.
2. These tweet-word co-occurrences could then be used to calculate the centroid of a given tweet based on the word vectors present in the tweet.
3. Once step 2 has been completed for all tweets for a user, the vector representing the user could then be generated by calculating the centroid of all tweets for the given user.

From the perspective of temporal analysis, steps 1-3 described above would be applied to each time bin for each user, resulting in a single vector for each time bin for each user. By comparing these vectors, it would then be possible to observe the evolution of the user over time based on the words they use.

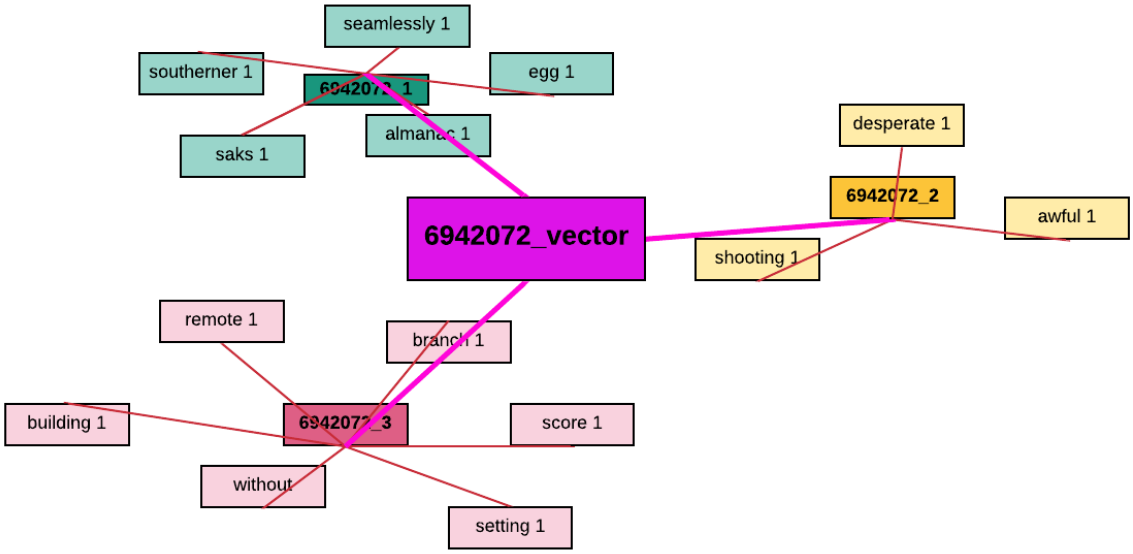

Figure 3-2: User Vector Generation
This image illustrates how the centre of each tweet is obtained and then used to compute the centre of the tweet vectors to generate the user vector.

More detail on the implementation and processes involved in developing the system are discussed in more detail in *Chapter 4*.

3.2.3 Non-Functional Considerations

The primary non-functional constraint identified during the research was concerned with the legal and ethical issues of research using user data obtained from Twitter. Despite the fact that there are several benefits to using datasets provided and obtained from Twitter, there are clearly many ethical and legal implications to be considered as was observed in *Section 2.1.3* of this document.

3.2.2.1 Collection and Processing of User Data

For the user data relevant to this research, data collection was not part of the process: Rather, the data had already been collected and made publicly available by Liang *et al.* as discussed in *Section 3.2.1.2*. In all sources found, Liang *et al.* did not state whether or not consent was obtained for the collection of the user data. Information regarding the data gathering process is limited to the following: “1,375 users randomly sampled from Twitter... 3.78 million tweets posted by the users from the beginning of their registrations up to May 31, 2015”. [30] [31]

[65] Thus, in order to ascertain that this dataset could ethically and legally be used as part of the research, it was decided that analysis of the GDPR regulation itself was required.

- In accordance with clause 39 of the GDPR, "...the specific purposes for which personal data are processed should be explicit and legitimate and determined at the time of the collection of the personal data".

- Additionally, clause 50 of the GDPR states that "the processing of personal data for purposes other than those for which the personal data were initially collected should be allowed only where the processing is compatible with the purposes for which the personal data were initially collected. In such a case, no legal basis separate from that which allowed the collection of the personal data is required."

With acknowledgement of the absence of legal expertise and guidance on the part of the author, these clauses appear to state that (i) consent must be obtained from the relevant persons at the time of collection of the data, and (ii) if additional processing is carried out on the data, there is no requirement to obtain additional consent from the relevant persons if it is being carried out for "compatible" purposes. When applied in the context of this research, it seems appropriate for the author to believe that (a) those who collected the data originally did so with appropriate user consent, and that (b) since the purposes of processing the dataset are no different to that of those who collected it initially, there is no additional consent required.

3.2.2.2 Ethical Use of User Profiling Research Results

Great emphasis was placed on clarifying the scope of the analysis to be conducted on the obtained user data such that ethical guidelines would not be breached by the results.

- Any results generated, in line with guidelines provided by Twitter Inc., will not be used to infer information about users such as their health, financial status or condition, political affiliation/beliefs, racial or ethnic origin, religious/philosophical affiliation/beliefs, sex life/sexual orientation, trade union membership, or alleged/actual commission of a crime. [65]

- Since the input Twitter user dataset is being used purely for metric-based evaluation of temporal embedding models and not to derive data insights for further use, user profiling inferences will not be evaluated in and of themselves or published as part of the findings of this research.

3.2.4 Evaluation Considerations

Substantial consideration was given to how the user profiling system would be accurately evaluated. There are a number of specific considerations in this regard, detailed below.

3.2.4.1 Evaluation Metrics

The primary measure of the effectiveness of any system is based on metrics. The decisions with regard to the generation of appropriate metrics were as follows:

1. **Relevance**
   It was concluded that the most appropriate way to measure the performance of the user profiling system, i.e. whether the words returned are the most closely associated to the given user, would be through use of binary relevance comparison to the ground truth data[51].
2. **Metric Generation**
   For the purposes of calculating relevance metrics for the system, it was decided that Trec Eval[52] would be used. This would enable a number of standard metrics to be produced, including precision, recall and mean-average precision.

More information on both the metrics and relevance computations are provided in *Appendix A.3*.

3.2.4.2 Baseline Comparison

With regards to evaluating TRI's performance in representing users as vectors, a baseline comparison is required. Given thatWord2Vec is one of the most popular embedding techniques at present and that Liang *et al.* used a temporal extension of this model for their research, Word2Vec was proposed as the most appropriate embedding technique for a baseline comparison to the TRI implementation. This means that the same setup for TRI is required to be employed for Word2Vec, i.e. it must be provided with the same input data, undergo the same processing, and have its output evaluated based on the same evaluation metrics.

[51] When using binary relevance, this comparison is as simple as determining whether the word associated with the user by the system is correct or not, i.e. whether or not it matches the corresponding ground truth value.
[52] Trec Eval is an industry-standard tool published by NIST, used to evaluate systems based on relevance judgments. [95]

- It was decided that the Word2Vec Skip-gram model could be implemented in Python, since the *gensim* library comes with this model built-in meaning that minimum effort and maximum implementation accuracy could be leveraged. [62]
- In order to keep the process of evaluation fair, the same processes applied to TRI would be applied to Word2Vec. To this end, it was decided that both the TRI and temporal Word2Vec models would be provided with the input dataset, and have their top-ten[53] most closely associated words to a user vector[54] returned for evaluation against the ground truth data.

In order to compare against this baseline, it was also decided that the dataset be compared for the several time bins similar to the work completed by Liang *et al.* In Basile *et al.*'s implementation of TRI, the system was designed only to process data in time bins of years; In contrast, Liang *et al.* evaluated their system for other time periods, such that *period* $\in$ *{week, month, quarter-year, semi-year, year}*[55]. Thus, it would be required to extend the TRI implementation to handle a variable time period from a finite set of possible time periods.

The experiments conducted, and the evaluation of the system are discussed in more detail in *Chapter 5*.

3.3 Challenges

A number of challenges were identified during the design of the temporal user profiling system. The manner in which these were addressed during the design phase is described in the following subsections.

3.3.1 Dataset Inherent Challenges

As described above in *Section 3.2.1.1*, it was identified that Twitter OSN data would be most suitable for the purposes of this research. Although there are many advantages to using OSN data, there are also some inherent disadvantages: One of the most quoted reasons being that

[53] This value was selected randomly as an initial value, with the intention that if time permitted the impact of varying this value could be evaluated.
[54] This set of vectors can be determined by applying the Cosine Similarity measure to each user vector for all word vectors in the time bin, and using the k-nearest neighbours approach to determine the top-ten most similar vectors.
[55] With regards to the Word2Vec implementation, a separate NN model would have been generated for each time bin for each time period.

there is a significant amount of noise present in such data. [67] Challenges with the UCL Twitter dataset identified at the design stage are described below.

3.3.1.1 Misspellings and Abbreviations

Since there is no auditing of the quality of content being posted by users on Twitter, tweets tend to contain many misspellings such as that shown in *Figure 3.5*. Twitter users are also restricted in the number of characters they can post at a given time[56] and thus also tend to use abbreviations where possible – for example, the use of 'IKR' instead of 'I Know Right'.

Figure 3-3: Spelling Mistake in Tweets
This image is an illustration of how easy it is for a user to produce a spelling mistake. In a tweet published by the US Dept. of Education, the name of NAACP co-founder W.E.B. Du Bois was misspelled. In the subsequent apology for the misspelling shown here, the US Dept. of Education produced yet another spelling mistake.

This poses issues for the use of Twitter data in research. Thus, it was decided that as part of the pre-processing of the dataset, any term which occurred only once in the dataset would be presumed to be a misspelling and thus be removed.

3.3.1.2 Emojis and Hashtags

There are a number of additional features of Twitter data which can enhance the semantic quality of the content if presented properly, but which can introduce substantial noise if not.

1. **Emojis**

   Emojis can enhance the semantic quality of tweets, since they represent emotions and reactions. However, in the UCL Twitter dataset these were found not to be correctly encoded in their Unicode representation, and instead were represented as either:

   - Punctuation characters i.e. ':)' instead of 'U+1F604' for the 😄 emoji; or

[56] In general, Tweets are limited to a maximum length of 280 characters.

- Incorrectly rendered characters, i.e. '' instead of 'U+1F604' for the emoji.

As a result of these issues with the dataset, emojis were determined to be outside of the scope of this research.

2. **Hashtags**

Twitter also offers a feature called a *hashtag*[57]. An example of which is shown in *Figure 3.6* These are loaded with semantic meaning since they are included in tweets to reflect a topic which is trending on the Twitter platform. For the purposes of this research, it was decided that these could be treated as being equivalent to a word in a user's vocabulary, and thus the only extra processing required on these was the removal of the preceding '#' character for each hashtag encountered.

Figure 3-4: Hashtags in a Tweet Example
This image illustrates a tweet with the hashtag '#EGGBOY'. This reflects a trending topic on Twitter in mid-March 2019, where a boy cracked an egg over a New Zealand politician's head. All tweets containing the '#EGGBOY' hashtag are listed on a single searchable webpage on the Twitter platform.

3.3.1.3 Languages in the Data

The presence of multiple languages in the UCL Twitter dataset poses additional problems, since there may not be sufficient language-specific data to learn how to best represent data

[57] A hashtag is used to index keywords or topics on Twitter, and allows users to easily follow topics they are interested in. Every hashtag is a continuous sequence of characters without whitespace separation, preceded by a '#'. By using this feature, hashtagged words that become very popular often become *trends*.

from a user. Users who tweet in multiple languages may even mix languages in a single tweet as in *Figure 3.7*, making them difficult to interpret from the perspective of research.

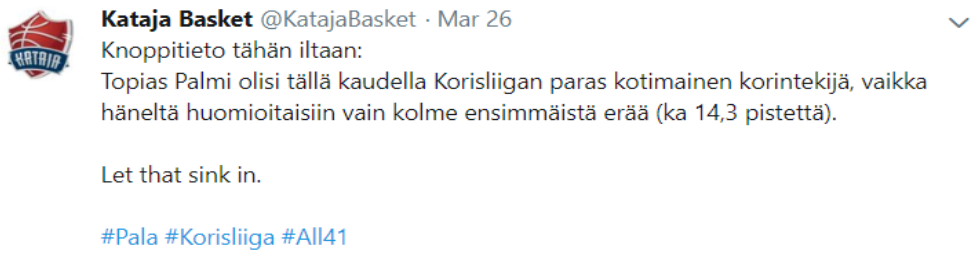

Figure 3-5: Bilingual Tweet Example
This image provides an illustration of how some bilingual users may use multiple languages in a single tweet; in this case, both Finnish and English are used in the same tweet.

Thus in order to simplify the analysis, it was decided that all languages except English would be considered outside of the scope of this research.

3.3.2 Data Enrichment

One of the most common issues that can occur in datasets from OSNs like Twitter is data sparsity. An example of a sparse tweet is shown below in *Figure 3.8*, demonstrating how it can be very difficult to understand the content and meaning of the tweet. The consequence of this is that sparse data can make generation of well-contextualised word vectors difficult.

Figure 3-6: Sparse Tweet Example
The above tweet illustrates the way in which tweets can be sparse. The tweet contains 2 words, 1 emoji and 1 hashtag - also happens to be a deliberate misspelling of the word WHAT.

It was identified that for sparse tweets in the UCL dataset, there is potential to enrich them using embedded URLs: By visiting the webpages of embedded URLS, the HTML data can be extracted and parsed to provide more context to the tweet of the given user[58].

3.4 Summary

A diagram of the final proposed system design can be seen diagrammatically below in *Figure 3.9*, illustrating how the identified components interact with each other to provide the overall user profiling system, starting with data pre-processing and database insertion to the use of TRI to generate the embeddings. The system in its entirety was designed such that it all issues foreseen prior to implementation were addressed to the fullest extent possible.

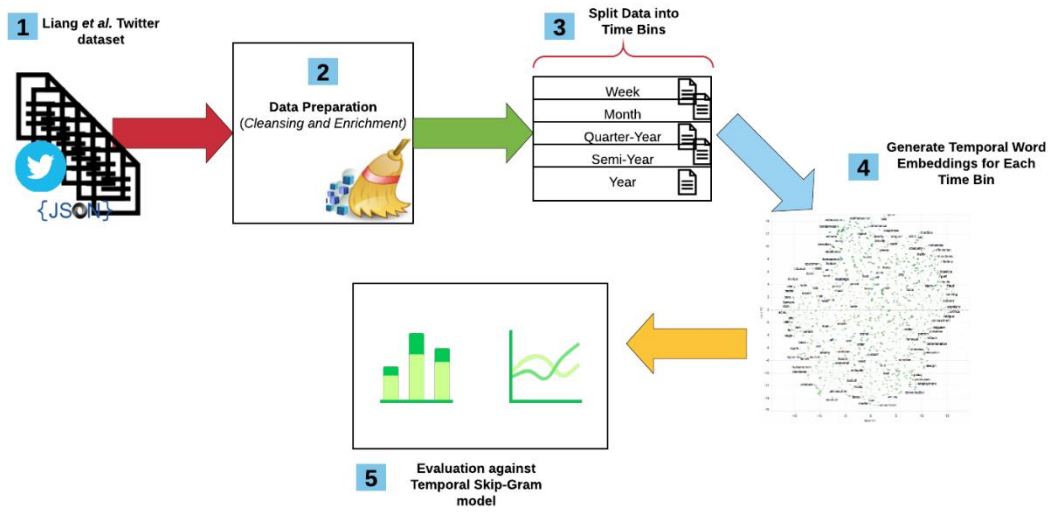

Figure 3-7: Overview of TRI System Implemented
This image is an illustration of the system whose design is described in this chapter, which uses TRI to generate temporal models of users using word embeddings.

[58] Many of the problems that are inherent in the tweets may also be applicable to the URLs: For example, the URL may link to a webpage that uses a non-English language. However, an assumption is made here that the number of such webpages will be negligible.

Chapter 4 Implementation

In this chapter, the implementation of the various components of the research system are described in depth. This includes the numerous challenges which were encountered during the implementation of the user profiling system, which was developed entirely in the Java programming language.

For the purpose of succinctness, low-level details of installation of tools etc. are generally not explained in great detail. Instead, attention is given to the implementation of the key components of the system, elaborating on the decisions that were made.

As motivated in *Chapter 3*, the implementation of TRI used in this research system leveraged a previously implemented version of TRI developed by P. Basile[59], who has completed several major research projects into the use of and effectiveness of TRI.

Section 4.1 discusses the initial processes performed upon dataset used in this system, including the implementation of preliminary pre-processing performed upon it before subsequently storing the refined data in an IMDB. The use of data enrichment by capitalising on the embedded URLs within tweets is also discussed in this section.

Section 4.2 discusses how the refined dataset further processed for use in the generation of user profiles, before explaining the process of implementing TRI. This includes an explanation of the construction of word-word and tweet-word co-occurrence matrices, before finally generating both word and user embeddings.

Finally, *Section 4.3* provides a brief summary of what was achieved as part of the implementation phase of the research.

4.1 Dataset Preparation and Storage

As discussed in *Chapter 3*, the dataset used in this research is a publicly available Twitter dataset which was originally collated by University College London's Big Data Institute[60]. Full statistics for this dataset are provided in *Appendix A.2* of this dissertation.

[59] P. Basile *et al.*'s open-source implementation of TRI is available on GitHub. https://github.com/pippokill/tri
[60] The UCL Twitter dataset is publicly available from S. Liang's BitBucket account.

4.1.1 Data Format

The dataset contains the data of 1,198 Twitter users[61]. The data pertaining to each individual user is stored in a distinct file.

- In every file, each line corresponds to a tweet or retweet posted by the given user, represented as flattened JSON objects.
- The JSON tweets contain many different fields. The subset of fields relevant to this research are shown in *Table 4.1*.

Field Name	**Description**	**Relevance**
user_id	The unique Twitter ID of the user.	Makes it possible to identify all of the tweets posted by a given user.
created_at	The timestamp of creation of the tweet, e.g. "Wed Oct 08 22:36:55 +0000 2014".	Makes it possible to perform temporal user profiling upon the data, since it is possible to identify when a tweet was created.
lang	The classified language of the tweet.	Contains the language's ISO 639-1 code, used to filter tweets by language in this research.
retweeted	A boolean flag that indicates whether a tweet object is a tweet or retweet.	Used to determine whether this tweet is a tweet or a retweet.
retweeted_status	If *retweeted* field has the value *'*true*'*, this field contains any tweets embedded within the tweet as retweets.	Contains the entire information relating to a retweet, which is equally as important as a tweet and contains all of the subfields of a normal tweet.
text	The content of the tweet, including text, embedded images, videos, and URLs.	Corresponds to the content of tweets and retweets, which is essential to this research as it is the predominant information used in the modelling of users.
entities	A list non-textual entities within the tweet such as URLs.	A sub-attribute of this is the *url* field, containing a list of the URLs present in the tweet, used in data enrichment in this research.

Table 4-1: Example Fields of Twitter Data Used
This table provides the identifiers and descriptions of the most relevant fields present in the JSON dataset.

[61] In their paper, Liang *et al.* report that the dataset contains 1,375 users data: However, from the same published dataset cited in their paper, [31] 1,198 users' data were actually found to be present. Thus, this is a discrepancy between the cited and actual number of users' data present in the dataset.

As described in *Section 3.3.2*, some of these fields may be sparse in content and context, an issue which is countered using data enrichment which is described below in *Section 4.1.2.3*.

4.1.2 Data Pre-Processing and Storage

In a project that relies exclusively on data, one hugely important step is the cleansing and preparation of the data for the system by applying measures such as text normalisation to remove any anomalous data to the maximum extent possible.

This section describes the implementation of preliminary data cleansing processes applied to the dataset before it could be ingested by the TRI user profiling system. This includes the implementation of a set of sequential logical conditions expressed in Java[62], as well as the eventual storage of the pre-processed data in a MongoDB database where it could be accessed on-demand thereafter.

4.1.2.1 Preliminary Data Pre-Processing

The first data pre-processing step is to read in each users' data from their respective data file line-by-line. For the purposes of preserving all potentially valuable information available in the data, not all data pre-processing required for the project was carried out at this stage[63]. The sequence of pre-processing steps that were carried out at this stage are described in order below.

1. **Language Filtering**

   As mentioned in *Section 3.3.1.3*, only tweets determined to be written in English are retained for analysis: All other tweets are omitted as it is not within the scope of the research to investigate embeddings across multiple languages.

   In total, 30 different languages were discovered within the dataset, a list of which can be found in *Appendix A.2*. The language of each tweet was determined by applying an OR operation to (i) the *lang* field of the tweet, and (ii) the language determined by the Apache Tika[64] Language Detector[65]. This operation can be seen in *Snippet 4.1*.

[62] All development was included as an extension to the open-sourced TRI application developed by Basile *et al.*, which formed the basis of the temporal user profiling system.
[63] The reason for this was to remove all redundant data from the start, storing only the necessary data - including any data that could be potentially used in further research.
[64] Apache Tika is a content detection and analysis framework which was used to extract text content from tweets, as well as analyse the languages and the metadata associated with each tweet. [76]
[65] It was decided to use the Apache Tika framework in this way in order to improve the accuracy of the language classification of the tweets: Any tweets misclassified by Twitter could potentially impact upon the quality of

```
LanguageDetector detector = new OptimaizeLangDetector().loadModels();
LanguageResult result = detector.detect(tweetText);

if(!jsonTweet.get("lang").toString().equals("en") && !result.getLanguage().equals("en")){
    nonEnTweet++;
    continue;
}
```

Snippet 4-1: Removal of Non-English Tweets
This snippet illustrates Java code corresponding to the OR operation applied to the lang field of the tweet object, and the language detected by Apache Tika. The code checks for the condition when both languages are not English, producing an OR operation

Although most non-English tweets were removed successfully, this process was not perfect: An example of a tweet primarily written in English which was classified as being written in Finnish is shown in *Figure 4.1*.

"#dinglefit final week! (@ Forever T��l� - @forevertoolo in Helsinki) https://t.co/GXMjnzpoh2"

Figure 4-1: Bi-Lingual Tweet from Dataset Example
This sample tweet illustration of how some tweets in the dataset which contain English are misclassified due to some words being from a different language.

Of the 3,658,673 tweets present in the dataset, a total of 3,039,118 were classified as being written in English and thus included for further analysis.

2. **Check for Presence of Alphanumeric Characters**

   A conditional check was implemented which filtered out any tweets which were determined not to contain any alphanumeric characters, since such tweets could clearly not be used to construct word embeddings. This filtering was achieved using a simple regular expression.

3. **URL Extraction**

   The next step in the data pre-processing is to extract the URLs that are embedded in the within the tweet. The purpose of this is twofold: (i) The URL itself should be removed from its respective tweet, since it cannot be used to generate word embeddings; and (ii) the URLs will be later visited to have their HTML content scraped for the purpose of data enrichment.

   Extraction of the URLs was achieved by accessing the *url* subfield in the *entities* field of the tweet, and storing all URLs found in a list. This list was then used to detect and

results obtained, since (i) tweets primarily written in English could potentially be omitted from the analysis, and likewise (ii) tweets primarily written in an alternative language could potentially be included in the analysis. The Apache Tika *OptimaizeLangDetector* function was used to aid in inferring the language. [77]

remove URLs present in the tweet *text* field, requiring use of the regular expression shown in *Snippet 4.2*.

```
public static String removeUrlsFromText(String text) {
	return text.replaceAll(
		"((https?|ftp|gopher|telnet|file):((//)|(\\\\))+[\\w\\d:#@%/;$(
		)~_?+-=\\\\.&]*)"
		, ""
	);

}
```

Snippet 4-2: Remove URLs from Tweet Text Example
This snippet illustrates the regular expression used to remove a URL from the text field of its corresponding tweet.

4. **Removal of Undesirable Text Content**

   The next processing step applied was the removal of any additional undesirable content from the tweets. This included:

   - Excess whitespace characters;
   - Diacritical marks;
   - Email addresses;
   - Punctuation marks, with the exception of certain occurrences of '#' and '@' characters[66];
   - Any words containing incorrectly rendered characters such as ''.

5. **Removal of Single-Word and Single-Character Tweets**

   The final text processing step applied was the removal of tweets that were either (i) a single character or (ii) a single word in length. The reason behind this decision was that such tweets have *no context*, and hence would provide no insights into the language used by a user: Revisiting the Distributional Hypothesis discussed in *Section 2.3.3.1*, "a word is characterized by the company it keeps". [68] Thus, since there is no "company" or context in a sentence of one word in length, it can be ignored.

Thus, after applying the above text processing to the tweet shown in *Figure 4.1* it appeared as shown in *Figure 4.2*.

[66] Any '#' or '@' characters which also had at least one letter immediately following them were not removed, since these would correspond to either (i) a mention of a twitter user or (ii) a hashtag. Removing these punctuation marks would remove the ability to identify tagged users and hashtags, which could potentially be investigated for their contextual value as future work.

"#dinglefit final week forever @forevertoolo in helsinki"

Figure 4-2: Preliminarily Cleaned Tweet Example
The above figure illustrates the resulting text after the preliminary text processing was applied to the text in Figure 4.1. It can be observed that (i) all punctuation is removed, (ii) all URLs are removed, (iii) all '@'s without an immediate character following are removed, and words containing incorrectly rendered characters are removed.

A high-level view of the data pre-processing components of the system described in this section is also shown in *Figure 4.3*.

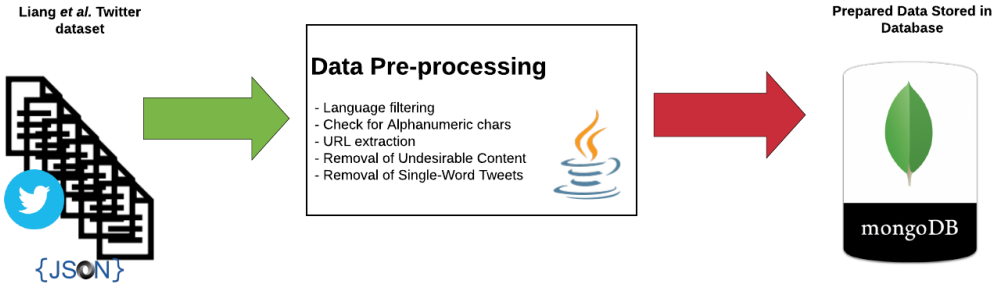

Figure 4-3: Diagram of Precleaning Process
A high-level view of the processes involved in the data pre-processing components of the user profiling system.

4.1.2.2 Data Storage

As described in *Section 3.2.2.1*, it was decided to store the pre-processed user data in a MongoDB IMDB. Standard installation and initialisation of the MongoDB instance was performed and made available within the Java code.

Before inserting the pre-processed data into the IMDB, the final step required to be applied to is to distinguish between whether each JSON object is (i) a tweet or (ii) a retweet, allowing them to be stored in separate lists in the user JSON object inserted into the IMDB[67].

- If the current tweet is identified to be a normal tweet, it is added to *tweet* list as part of the user object.
- If the current tweet is identified to be a retweet, it is added to a separate *retweet* list as part of the user object.

The final JSON object being inserted into the IMDB would hence look similar to *Figure 4.4*.

[67] The distinction between tweets and retweets is of relevance later in the development of the system, since retweets are of relevance to the generation of the *dictionary* and *vocabulary* in the construction of the word-word co-occurrence matrices for each time bin. However, retweets are not relevant to the generation of the tweet-word co-occurrence matrices for each time bin, since the language used in a retweet was chosen by a different user and is not necessarily the language which the user would choose to express the same sentiment.

```
[{ "_id" : "1",
   "twitter_id", "1503",
   "tweetList": ["tweet_1": {
                    "text": "Sample tweet 1",
                    "created_at": "Tue May 19 00:34:07 GMT 2015",
                    "URLs": {
                        "url_1": "https://t.co/pWNYfdiYRO"
                     }
                 },
                "tweet_2": {
                    "text": "Sample tweet 2",
                    "created_at": "Tue Mar 31 02:10:05 GMT 2015",
                    "URLs": {
                        "url_1": "https://t.co/9FiFabmXBy"
                     }
                 }],
 },
 …

]
```

Figure 4-4: Document Structure in MongoDB Example
Each user inserted into the DB will have (i) an ID, (i) a twitter_id, (iii) a list of tweets, where each tweet is comprised of a text field, data field, and a field to store a list of URLs from the tweet, and (iv) a retweet list, where each retweet has the same fields as the tweets.

4.1.2.3 Data Enrichment

The last data pre-processing step performed is the enrichment of the data stored in the IMDB: As described in *Section 3.3.2*, the use of HTML data scraped from URLs embedded in tweets can combat the issue of data sparsity.

A total of 848,560 URLs were obtained through the data pre-processing discussed in *Section 4.1.2.1*. The scraping of HTML data from these URLs was achieved as follows:

- **Classification of URLs as Active and Inactive**

  Each URL was sent a HTTP connection request. If a HTTP 200 response code was received[68], it was deemed to be active and stored in an active list to be scraped later. Otherwise the URL was deemed to be inactive, added to an inactive list, and no

[68] The reason as to why only the response code of 200 was considered was due to the other response codes not necessarily ensure the URL is active and able to have content retrieved from it.

longer considered for analysis[69]. The total number of URLs determined to be active was 48,420[70].

- **HTML Extraction from Active URLs**

  The list of active URLs was used for the scraping of HTML to enrich the dataset. For each URL, the HTML webpage content was retrieved by sending a HTTP GET request, and Apache Tika's *AutoDetectParser* was used to extract the usable content from the returned HTML response[71].

  After the HTML content had been parsed using Apache Tika, the same pre-processing steps which were applied to the tweet data in *Section 4.1.2.1* were also applied to the HTML data

The enriched tweets[72] are later used in the TRI process when generating the word-word and tweet-word co-occurrence matrices.

4.2 User and Word Embedding Generation

Once the cleaned data is present in MongoDB and enriched with URL content, the next phase is the generation of the word-word and tweet-word co-occurrence matrices. In simple terms, the output of this phase is a series of files, whose contents represent the number of times a word occurs within the same context window as the current word.

Two of the following three subsections, *Section 4.2.2.1.1* and *Section 4.2.2.1.2*, relied heavily on the prior work done by Basile *et al.* in their open-sourced implementation of TRI. [69]

4.2.1 Data Cleansing

Prior to the generation of any word embeddings, additional text processing was required in order to generate co-occurrence matrices suitable for the purposes of this research. Besides

[69] In these files, the user's ID and timestamp associating the URL to a given tweet were included. The inactive URLs were recorded with the intention that if time permitted, they could be analysed by checking for a historical record of their content using a tool such as the WaybackMachine API. [92]

[70] This number corresponds to approximately 5.7% of the total number of URLs extracted. This is a symptom of the age of some of the tweets included in the dataset, meaning that some of the URLs were no longer active or had relocated.

[71] This method is defined as part of the Apache Tika framework, automatically detecting the format of the input content and applying basic text parsing to extract it.

[72] Though not all tweets are enriched since they either (i) never contained a URL or (ii) their embedding URL(s) could not be scraped for data, the term *enriched tweets* refers to the entire set of tweets after enrichment has occurred.

lemmatisation which is applied wholly before the generation of co-occurrence matrices, the steps described in this section are applied to each enriched tweet one-by-one during the generation of the co-occurrence matrices, which is the subject of *Section 4.2.2.1*.

4.2.1.1 Removal of Misspellings and Stop Words

Stop words were removed from the data, as well as words which occurred very infrequently within the dataset: The reasoning behind this being that words which occur very infrequently are highly likely to be spelling mistakes[73].

For stop word removal, a list of English stop words obtained from xpo6.com was used. [70] For removal of misspellings, a *blacklist* was constructed:

- The frequency of occurrence of each word in the dataset was measured and stored in a map data structure, where the key corresponded to the given word and the value corresponded to the integer frequency of occurrence of the word. Each time a given word occurred in the dataset, its frequency in the map was incremented by 1.
- If the word only occurred in the corpus twice or less, it was assumed to be a non-word and was added to the *blacklist* which acts as a stop word list for the dataset.

Both the stop word list and the blacklist are later used during the process of generating the co-occurrence matrices described in *Section 4.2.2.1*.

4.2.1.2 Lemmatisation

The next data cleansing process applied to the enriched tweets is lemmatisation, for which the StanfordCore NLP library was used[74] .

- The tweet text is passed to the lemmatisation function, where it is iterated across and split into individual tokens.

[73] However likely this is, it can lead to the removal of important words which occur infrequently in the dataset and which are not misspellings. This issue was recognised during the implementation, and one attempt to improve such mislabelling of words as non-words was to use an API call to a dictionary such as the one provided by Oxford Dictionaries. However, upon investigation it was found that to use these APIs required a financial payment and hence, this option could not be pursued.

[74] StanfordNLP is a publicly available NLP library maintained by Stanford University. As described on their site, "it provides a set of natural language analysis tools which can take raw English language text input and give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, and mark up the structure of sentences in terms of phrases and word dependencies, and indicate which noun phrases refer to the same entities." [78]

- These tokens are in turn iterated across, and passed to the lemmatising function word-by-word.
- The lemmatised words are then finally concatenated back into a single string to give the lemmatised tweet.

4.2.1.3 Abnormal Punctuation Removal

It was discovered during this phase that there were a number of non-standard punctuation characters which had not been successfully filtered out of the dataset in prior data pre-processing stages, such as the "`" character. Each time one such abnormal character was identified, the offending character was added to the *blacklist* mentioned in *Section 4.2.1.1* to remove it from the tweets.

4.2.2 Temporal Random Indexing

As discussed in the *Chapter 3*, TRI has a high potential for use in the generation of user embeddings to model a user. A detailed explanation of the stages involved in RI and TRI can be found in *Section A.4.1* and *Section A.4.2* respectively. It is advisable to review the contents of these sections, since the focus here is on the practical implementation of the TRI-based user profiling system rather than on theory.

It should be noted that unless otherwise stated, each step described here corresponds to a process applied to the data for a single time bin. As a temporal user profiling system, in reality these processes are repeated in their entirety for each time bin being considered.

4.2.2.1 Generating Co-Occurrence Values

As described in *Appendix A.4*, TRI involves the generation of word-word co-occurrence matrices, which essentially capture the number of times each word occurs in the same context as the current word. In order to allow for a user vector to be generated, the TRI process is extended to compute tweet-word co-occurrence matrices[75], which in turn are used to compute a user vector as described in *Section 4.2.2.2*.

[75] Note that the terms 'tweet-word embedding'/'tweet-word co-occurrence matrix' and 'user embedding'/'user co-occurrence matrix' are used interchangeably throughout this section and can be taken to have equivalent meaning.

The following section describes the implementation work involved in generating the word-word and tweet-word co-occurrence matrices for this research. The basic components of the word-word co-occurrence matrix generation described in this section had previously been implemented by Basile *et al.*, significantly reducing the development work required. However, the development carried out in this component of the system was concerned with extending the existing implementation such that:

1. The data cleansing steps described in *Section 4.2.1* would be applied to all content before being used to generate word embeddings;
2. The system would be capable of consuming JSON data from Twitter rather than news article data from text files;
3. The system would be capable of generating temporal word embeddings for multiple different time periods rather than only for years.

All components of the tweet-word co-occurrence matrix generation and the subsequent user embedding generation were developed in their entirety as part of this research.

4.2.2.1.1 Constructing the Dictionary and Vocabulary for a Given Time Bin

The first step in the generation of co-occurrence matrices is the generation of a *dictionary* and a *vocabulary* for the given time bin. This involves first iterating across the corpus and for each time bin, accumulating the enriched tweet associated with all users in the dataset. Once accumulated, the resulting data looks similar to that shown in *Figure 4.5*, where all of the enriched tweet text from all users associated within each time bin is tokenised, and the resulting tokens for that time bin stored in a list.

Week 1 = [contents, from, tweet, one, week, one, contents, from, tweet, two, week, one, …]
Week 2 = [contents, from, tweet, one, week, two, contents, from, tweet, two, week, two, …]
…
Week N = [contents, from, tweet, one, week, N, contents, from, tweet, two, week, N, …]

Figure 4-5: Example of Aggregated Text

This figure provides An illustration of the way in which the enriched tweets are first aggregated for all users for a given time bin (e.g. week 1), and then are tokenised and added to a list.

Next, a distinct *dictionary* is generated for each time bin[76]. The purpose of the dictionary is to record which words are the most frequently used in the given time bin. An illustration of how the dictionary is constructed is shown in *Figure 4.6.*

[76] For example, week 1 will have its own dictionary, as will week 2, year 1, year 2, etc.

- First, the data cleansing processes described in *Section 4.2.1* are applied to the list for the given time bin.
- After the data cleansing, the remaining tokens are iterated through one-by-one.
    - Each time a token is encountered for the first time, it is added as an entry in a hashmap which represents the dictionary. The key of the hashmap is the token itself, and the value is the number of occurrences of that token encountered.
    - Each time a token which already exists in the hashmap is encountered, its hashmap entry is updated by incrementing the number of occurrences encountered.
- Finally, the dictionary is sorted by descending order of occurrence of the words. The dictionary is then truncated such that the top 50,000 words are retained, whilst the remainder are discarded[77].

Week 1 (Tokenised List) = [dogs, are, great, cats, are, better]

Week 1 Dictionary = {
<dogs, 1>,
<are, 2>,
<great, 1>,
<cats, 1>,
<better, 1>
}

Figure 4-6: Generated Dictionary Example
The above illustrated shows how, for time bin 'week 1'. Here, the tweet tokens are 'dogs', 'are', 'great', 'cats', and 'better', where 'are' is the only token which occurs more than once.

After the dictionary for the time bin has been constructed, the *vocabulary* for the time bin is constructed. This is simply a list which records the tokens in the dictionary without recording the number of times that they occur in the time bin, as shown in *Figure 4.7*.

[77] This value is the same as that chosen by Basile *et al.* in their implementation of TRI. Although there was not sufficient time available to experiment with this value, it is plausible that adjusting this value to be either larger or smaller may have changed the final user vectors generated.

Week 1 (Tokenised List) = [dogs, are, great, cats, are, better]

Week 1 Dictionary =
{
<dogs, 1>,
<are, 2>,
<great, 1>,
<cats, 1>,
<better, 1>
}

Week 1 Vocabulary =
{
dogs,
are,
great,
cats,
better
}

Figure 4-7: Visual Representation of Dictionary and Vocabulary Generated
An example of a vocabulary generated based on the dictionary shown in Figure 4.6 for time bin 'week 1'. It can be observed that the same tokens are recorded in both, but that the vocabulary does not store a corresponding frequency of occurrence.

4.2.2.1.2 Generating the Word-Word Co-Occurrence Matrix for a Given Time Bin

Once the dictionary and vocabulary have been generated, the word-word co-occurrence matrix is generated. This is achieved by iterating through the tokenised list for the current time bin, checking whether the current token exists in the vocabulary.

- A context window is applied across the context of the word in the tokenised list[78]. The high-level objective of this exercise is to count the number of times that any word occurs within the same context window as the current word. The result of applying this process across the entire tokenised list is the generation of the co-occurrence matrix, an example of which is shown in *Figure 4.8*.

[78] A context window of size 3 words was chosen at random as an initial value. Due to time limitations, the effects of adjusting this value were not investigated: However, this would be an interesting parameter to look into further. Very large context windows can provide too much context to be useful, whilst smaller ones provide less: Striking the right balance to achieve the best results is likely to be a challenging undertaking.

	dogs	are	great	but	i	prefer	cats
dogs	0	1	1	0	0	0	0
are	1	0	1	0	0	0	0
great	0	1	0	1	0	0	0
but	0	0	1	0	1	0	0
i	0	0	0	1	0	1	0
prefer	0	0	0	0	1	0	1
cats	0	0	0	0	1	1	0

Figure 4-8: Word-Word Co-Occurrence Matrix Example
This is a visual representation of a simple word-word co-occurrence matrix for the token list [dogs, are, great, but, i, prefer, cats] for a context window of size 3. It can be seen for instance that the word 'cats' co-occurs once with 'i' and once with 'prefer'.

- In order to generate the word-word co-occurrence matrix, a temporary record of the position of each token is generated, such that each token can be mapped to its index position in the tokenised list. The generated co-occurrence matrix is stored as a nested hashmap, an example of which can be seen in *Figure 4.9*.
- The key of the outer hashmap is the index position of the token to which the context window is being applied.
    - The value of the outer hashmap is also a hashmap, where the key corresponds to the current co-occurring word being considered in the context window.
    - The value of the inner hashmap corresponds to is a map which contains the integer index position of the words in the same context as the current key along with the number of times the word has occurred in the same context.

This word-word co-occurrence matrix is written to a text file, which is named according to the time bin for which it was generated.

Week 1 (Tokenised List) = [dogs, are, great, but, i, prefer, cats]

index('dogs') = 0
index('are') = 1
index('great') = 2
index('but') = 3
index('i') = 4
index('prefer') = 5
index('cats') = 6

Week 1 Co-Occurrence Matrix =

```
{
   <0, {
      <1, 1>, <2, 1>
     }>
   <1, {
      <0, 1>, <2, 1>
     }>,
   <2, {
      <1, 1>, <3, 1>,
   }>,
   <3, {
      <2, 1>, <4, 1>
     }>,
   <4, {
      <3, 1>, <5, 1>
     }>,
   <5, {
      <4, 1>, <6, 1>
     }>
   <6, {
     <4, 1>,  <5, 1>
   }>
}
```

Figure 4-9: Visual Representation of Word-Word Co-Occurrences Data Structure

This is an illustration of a basic word-word co-occurrence matrix in the form of a nested hashmap. For example, the index 0 corresponds to the index position of the word 'dogs', which has 'are' and 'great' occur in the same context once each

The entire word-word co-occurrence process is repeated for each time bin, and each is later used in the generation of the user vectors. A more general representation of the word-word co-occurrence matrix would look similar to that shown in *Figure 4.10*.

word_1 =	{ $word_{00}$*N_{00} ,	$word_{01}$*N_{01},	⋯	$word_{0k}$*N_{0k}}
word_2 =	{ $word_{10}$*N_{10}	$word_{11}$*N_{11},	⋯	$word_{1k}$*N_{1k}}
⋮	⋮	⋮	⋱	⋮
word_N =	{ $word_{n0}$*N_{n0}	$word_{n1}$*N_{n1},	⋯	$word_{nk}$*N_{nk}}

Figure 4-10: Visualised Representation of Word-Word Co-Occurrence Matrix

This provides a generic representation of the structure of the word-word co-occurrence matrix. In the above illustration, $word_{i\,j}$ represents each unique word that occurs in the same context as word_N, and $N_{i\,j}$ represents the number of times each unique word occurs in the given time bin.

4.2.2.1.3 Generating the Tweet-Word Co-Occurrence Matrix for a Given Time Bin

Once the word-word co-occurrence matrix has been generated for a given time bin, the next step is to generate the tweet-word co-occurrence matrix for that time bin. Tweet-word co-occurrences are generated for each tweet a user has posted, and correspond to the number of times each word in the vocabulary for that time bin occurs in a given tweet.

- For the given time bin, the vocabulary described in *Section 4.2.2.1.1* is consulted.
- For each user who posted tweets in that time bin, each tweet is iterated through token-by-token.
    - If the current token in the tweet exists in the vocabulary, it is kept and its frequency of occurrence within the tweet is recorded.
    - Otherwise, if the current token in the tweet does not exist in the vocabulary, it is discarded.
- Once the entire tweet has been iterated through, any surviving tokens and their frequency of occurrence are written out to a text file, where they are stored alongside a unique identifier for the tweet[79].

When all tweets have been iterated through for all users, the text file will consist of multiple rows of co-occurrence values as shown in *Figure 4.11*. Each row represents the number of times that each of the vocabulary words occurs in the given tweet.

	southerner	seamlessly	saks	almanac	desperate	shooting	awful	.....	.....	bloomberg	even	reason	list	austin	sox	enema
6942072_1	1	1	1	1	0	0	0	...	...	0	0	0	0	0	0	0
6942072_2	0	1	0	0	1	1	2	...	...	0	0	0	0	0	0	0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9654123_99	0	0	0	0	0	1	0	...	...	1	3	2	1	0	0	0
9654123_100	0	0	0	1	0	0	0	...	...	0	0	0	1	3	1	1

Figure 4-11: Tweet-Word Co-Occurrence Matrix Example
This graphic provides an illustration by example of the tweet-word co-occurrence matrix for a specific time bin. Each row corresponds to a tweet (identified by <user_id>_<tweet_number>), each column corresponds to a word in the vocabulary for the time bin, and each cell corresponds to the number of co-occurrences of the word in the given tweet.

[79] This unique identifier is formulated as the concatenation of (i) the *user_id* for the tweet, and (ii) the number of occurrence of the tweet for that user (based on when it occurred in time), separated by an underscore. For example, for the 1st tweet posted by user 1234567 for time bin 'week 1', the tweet identifier would be '1234567_1'.

Taken all together, this file represents the entire tweet-word co-occurrence matrix for the given time bin. A more generic form of this matrix is shown in *Figure 4.12*.

User1_1 =	{ $word_{00}*N_{00}$,	$word_{01}*N_{01}$,	⋯	$word_{0k}*N_{0j}$}
User1_2 =	{ $word_{10}*N_{10}$	$word_{11}*N_{11}$,	⋯	$word_{1k}*N_{1j}$}
⋮	⋮	⋮	⋱	⋮
UserP_Q =	{ $word_{i0}*N_{i0}$	$word_{i1}*N_{i1}$,	⋯	$word_{ik}*N_{ij}$}

Figure 4-12: Visualised Representation of Tweet-Word Co-Occurrence Matrix
This graphic provides A generic expression of the tweet-word co-occurrence matrix for a given time bin, for P users and Q tweets per user, where N is the frequency of occurrence of $word_{ij}$, i is the number of tweets in the time bin, and j is the number of words which exist in both the tweets and the vocabulary.

For the given time bin, each user can be represented as a sub-matrix of this overall co-occurrence matrix, by extracting each row of the matrix which corresponds to their *user_id*. An example can be seen in *Figure 4.13.*

	southerner	seamlessly	saks	almanac	desperate	shooting	awful	stomp	rehearsal	lawyer	arsenal	hair	official	beer	lap
6942072_1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
6942072_2	0	1	0	0	1	1	2	0	0	0	0	0	0	0	0
6942072_3	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1
6942072_4	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0
6942072_5	0	0	0	0	0	0	0	0	0	0	5	0	1	3	1

Figure 4-13: Tweet-Word Co-Occurrence Matrix
This graphic provides An illustration by example of a user co-occurrence matrix for a specific time bin. The user ID in this case is '6942072', and for this time bin they have posted 5 tweets. As in Figure 4.11, each row corresponds to a tweet (identified by <user_id>_<tweet_number>), each column corresponds to a word in the vocabulary for the time bin, and each cell corresponds to the number of co-occurrences of the word in the given tweet.

The tweet-word co-occurrence matrix can be converted into a single vector to represent a user by summing the vectors generated for each row in this matrix. This process is discussed in more detail in the following section.

4.2.2.2 Generating Word and User Embeddings

At this stage, both the tweet-word and word-word co-occurrence matrices have been generated for all of the time bins in the dataset. The next step is to generate vector representations for both the words and the users, i.e. the word embeddings and user embeddings.

As described in *Appendix A.4*, there are a number of stages involved in the generation of vectors using RI: (i) the generation of random Index Vectors (**IVs**) for each word in the vocabulary, and (ii) the generation of Context Vectors (**CVs**) for each word and tweet using the IVs, word-word co-occurrence matrix, and tweet-word co-occurrence matrix.

4.2.2.2.1 Generation of the Vector Dictionary

The first step in the generation of the vectors is the construction of a *vector dictionary*. Unlike the dictionary described in *Section 4.2.2.1.1*, the vector dictionary is not specific to any time bin and instead includes words from all time bins.

To construct the vector dictionary, the word-word co-occurrence matrices generated as described in *Section 4.2.2.1.2* are first read in from their respective files.

- Each row of each co-occurrence matrix is read in one-by-one. The first word from each line is extracted, which corresponds to the word whose context is described by that row of the matrix.
- For the word extracted from each row, its total context value is computed by summing the values of the co-occurrences in that row according to *Equation 4.1*, and stored in a map. It is this map which forms the dictionary.

$$w_i = \sum_{c \in C}^{N} c$$

Equation 4-1: Context Vector Function Definition
This equation : Illustrates how the context value of word w_i is generated, where c is the number of times a word occurs in the context C. The context value is recorded as an entry in the vector dictionary.

The above steps are repeated for each row of each word-word co-occurrence matrix, across all time bins. A conceptual view of this process is shown in *Figure 4.13*.

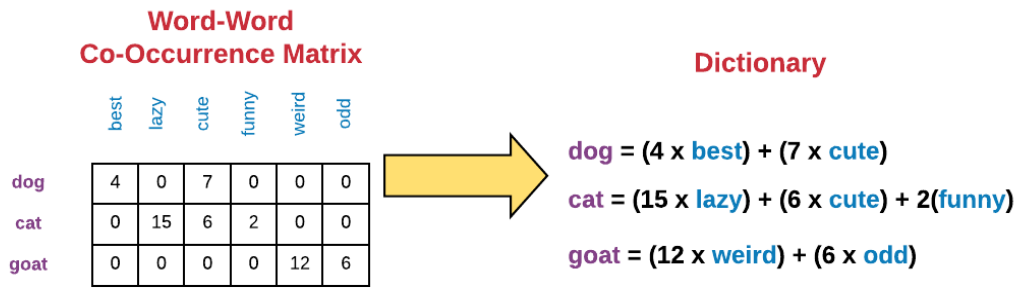

Figure 4-14: Visual Dictionary Generation
This graphic provides An illustration of the construction of the vector dictionary from a single word-word co-occurrence matrix. In reality, multiple word-word co-occurrence matrices, each corresponding to a different time bin, are used to generate the dictionary.

Once the full vector dictionary has been generated, it is sorted such that the highest context values are at the top and the lowest at the bottom.

4.2.2.2.2 Index Vector Generation

The next step is to generate a random IV for each word in the dictionary, which will in turn be used to generate the CVs for each word in the co-occurrences files. To generate the IVs, the vector dictionary is traversed, and for each entry in the dictionary a random IV is created and stored[80]. As described by Sahlgren, each IV generated is "real valued, ternary, and sparsely populated", where the majority of values are 0, and some values are either 1 or -1. [39]

4.2.2.2.3 Word Embedding Generation

One of the final phases of the implementation is the generation of the word embeddings, or context vectors, for each time bin. These vectors are constructed by iterating through the word-word co-occurrence matrix.

To generate an embedding for each word, the word-word co-occurrence matrix is read in row-by-row. First, the context vector for the word is initialised. It is thereafter defined by the sum of the contributions of each co-occurring word, which are computed one at a time and added to a running total.

- Taking each co-occurring word one-at-a-time, the IV for the current co-occurring word is retrieved from the vector dictionary.

[80] There was no additional implementation required to generate these random IVs: The existing implementation by Basile *et al.* leveraged an open-source package called *SemanticVectors* to generate the random IVs. [79]

- The frequency of co-occurrence of the current word is extracted from the appropriate cell of the co-occurrence matrix.
- Finally, the context vector is updated to include the contribution of the co-occurring word, which is defined according to *Equation A-5*.

This process is repeated for each word that co-occurs in the same context as the current word, producing the context vector for that word. The entire process is then repeated for every row of the co-occurrence matrix in order to generate a context vector for each word.

4.2.2.2.4 User Embedding Generation

One of the most important phases of the implementation was that of generating the user embeddings i.e. user vectors for each time bin using TRI, which is the most novel element of this research. Generation of these user embeddings is based primarily on the tweet-word co-occurrence matrix for the time bin[81].

The processes involved in generating the user vector are very similar to the processes involved in generating a word vector.

- As illustrated in *Figure 4.14*, for each tweet in the co-occurrence matrix, a *tweet vector* is generated by computing the centroid of the word embeddings corresponding to the co-occurring words present in the tweet-word co-occurrence matrix[82]. The centroid is computed as the mean of the contributions of the word embeddings[83]. This process is carried out for each tweet generated by each user for the current tweet-word co-occurrence matrix. The output of this is a set of tweet vectors for each user for the given time bin.

[81] The fact that the word embeddings were generated before the user embeddings is important, since the generation of the user embeddings is dependent on the prior generation of the word embeddings.

[82] For each word embedding used in this computation, the corresponding co-occurrence value, i.e. the value in the tweet-word intersection, is multiplied with its corresponding word embedding vector as a scaling factor prior to computing the centroid.

[83] For the purposes of this research, computing the centroid as the mean of the scaled contributions of the word embedding vectors was deemed sufficient. However, there is potential to explore other approaches for effectiveness in this regard as future work.

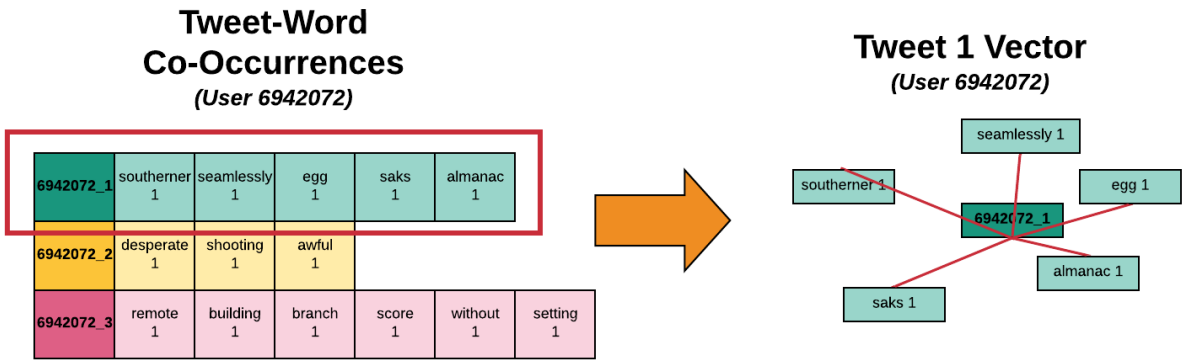

Figure 4-15: Tweet Vector Generation

This image provides An illustration of the generation of a tweet vector by computing the centroid of the word embeddings that co-occur with the tweet. The example shown here demonstrates the generation of a single tweet vector for tweet '6942072_1', where the contribution of each word is obtained by multiplying the word's embedding by its corresponding co-occurrence frequency, and the final tweet vector is computed as the statistical mean of the contributions.

- The final step required to generate the user vector for the current time bin is to compute the centroid of the tweet vectors for that user. The result is a unique user vector for the given time bin. This process can be seen visually in *Figure 4.15*.

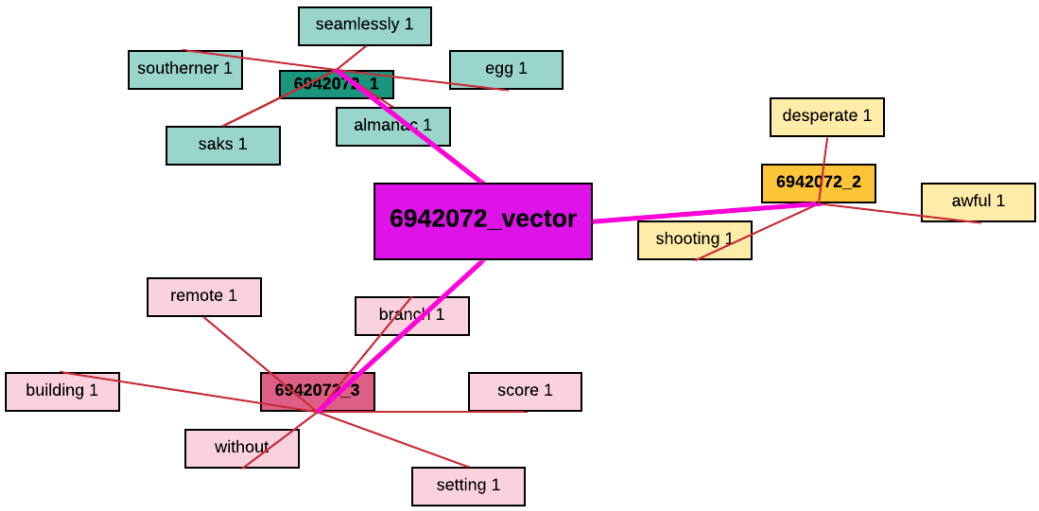

Figure 4-16: Representation of User Vector Generation

This graphic provides An illustration of the generation of a user vector for a given time bin, by computing the centroid tweet vectors for the tweets posted by that user within the time bin. The example shown demonstrates the generation of a vector for user 6942072 as the centroid of the vectors for tweets '6942072_1', '6942072_2', and '6942072_3'.

This generation of user vectors is repeated for each user for each time bin, resulting in a vector being generated for every user who tweeted in each time bin. All of the generated user vectors are written out to text files corresponding to their respective time bins, to be later analysed as part of the evaluation section of this research. Ultimately, it is the generation of these user vectors across multiple time bins which hold the potential to conduct temporal user profiling.

4.3 Summary

By the time each of the components of the system had been implemented as described in this chapter, an end-to-end user profiling system was in place. In this system:

1. Twitter user data is consumed from text files, pre-processed and enriched, and then stored in a MongoDB IMDB.

2. The data from the IMDB is then queried as-required, and subsequently cleaned by applying lemmatisation and removing stop words, misspellings and abnormal punctuation characters.

3. Word-word co-occurrence matrices and tweet-word co-occurrences matrices corresponding to time bins over time periods of weeks, months, quarter-years, semi-years and years are generated based on the cleansed data.

4. Finally, the word-word and tweet-word co-occurrence matrices are used to generate word embeddings, tweet vectors and finally user vectors.

Thus, the desired system to consume Twitter data and generating unique vector representations of both words and users in the same vector space had been delivered at this point. It is at this stage that the system is ready to be evaluated, with a focus on determining whether the the use of TRI is effective in generating temporal embeddings of users for the purposes of modelling their interests through time. This evaluation is the subject of *Chapter 5*.

Chapter 5 Evaluation and Results

This chapter provides an evaluation of TRI user profiling model implemented as described in *Chapter 4*, which includes a discussion of the design and execution of experiments carried out.

Section 5.1 discusses the ground truths required for the system, how the experimental setup was orchestrated and how the experiments were conducted, highlighting the specific tools and formats of the data required for the evaluation.

Section 5.2 provides the results obtained during the experimentation phase of this research.

Section 5.3 provides an in depth discussion into the results obtained during the experimentation phase of this research, and explicitly addresses the objectives set out when devising the research question for this dissertation.

Finally, *Section 5.4* provides a brief summary of the chapter.

5.1 Evaluation of Temporal Random Indexing

The evaluation phase of this research is focused on measuring how well TRI performs in modelling a user and their interests through time with respect to the vocabulary they use.

In any system evaluation, a baseline is required for comparison. As described in *Section 3.2.4.2*, the baseline implementation to be used in the evaluation of the TRI user profiling system is the Word2Vec Skip-gram model, which was applied to similar research by Liang *et al*. [31] Both the implementation and the evaluation of this model with respect to TRI are described in the following subsections.

5.1.1 Experimental Data

As data-driven research, a comprehensive evaluation of the implemented TRI user profiling system requires two sets of data:

1. A dataset used to generate vector representations of words; and

2. A ground-truth dataset, which is used as a "gold standard" against which the output of the system can be measured and conclusions drawn about its effectiveness[84].

For the purposes of this research, it was determined that **R**elevance-Oriented Ground Truth (**RGT**) should be used to evaluate the model: This would enable the determination of whether the TRI user profiling system capable of effectively inferring a user's interests or not. This decision was explained in *Section 3.2.4.1*.

5.1.1.1 Investigation of Existing Ground Truth Data

During the initial stages of this research, it had been identified that Liang *et al*. had made the ground truth data corresponding to the UCL Big Data Institute's Twitter dataset publicly available.

However, upon subsequent inspection of the ground truth data, it was realised that the data related to a different research dataset relevant to Liang *et al.*'s research into temporal user cluster modelling; i.e. for each time bin the users were clustered into cohorts based on inferred user interests. It was decided at this point to make contact with S. Liang, who had worked on multiple projects using this dataset. Unfortunately however, S. Liang explained in his response that this would not be possible:

> *"In terms of the labelling for users' profiles, a chinese company hired people to label the data for ground truth. According to the chinese company's rules as well as the Twitter company's rules (Twitter doesn't allow to share users' private information, see https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases.html: "You should be careful about using Twitter data to derive or infer potentially sensitive characteristics about Twitter users."), we can't make the profiling data redistribute to the public. Sorry for inconvenience.* ***You need to build your own ground truth****."*

Based on this recommendation, it was necessary to begin generating a custom ground truth dataset.

[84] In their paper, M. J. Tear et al. (2010) illustrate the importance of ground truths with their hypothetical question: "How can we be sure that the crime scene evidence does in fact belong to a convicted suspect?". [90] This simple example motivates the need for ground truth data, in that it isn't possible to validate an inference without having a comparison to the actual results.

5.1.1.2 Generation of a Ground Truth Dataset

With limited time remaining within which to complete the research, it was decided that the most effective means of generating ground truths was to annotate the existing data using the LDA algorithm. Although the use of this algorithm to generate the ground truth data would be highly likely to introduce bias since the ground truths would be generated based on the existing dataset, it would allow the implemented system to be evaluated to some degree.

Thus, the LDA algorithm was applied to the combination of cleansed tweet, retweet and scraped HTML data to generate five sets of ground truths: One for each of the five time periods considered in the system, i.e. yearly, semi-yearly, quarter-yearly, monthly and weekly.

- For each given Twitter user present in the dataset, LDA was employed to extract the top 10 words most relevant to the text associated to a given user.
- The words per user were extracted by keeping the proportion of each topic probability for a given tweet in the word generation[85].
- In order to measure the 'goodness' of each topic uncovered by the LDA algorithm, the *u_mass coherence* was used[86]. [70]

At the end, the best model was obtained with 50 topics, using Inverse Document Frequency (**IDF**) to weight the terms, which resulted in a coherence score of 0.603787265237163. Thus, a full set of ground truths was now available to perform an evaluation of the TRI user profiling system.

5.1.2 Experimental Design

The experiments conducted in this research are based upon the sole objective of answering the research question that was formulated in *Section 3.1.2* of this document: To determine the extent to which a Twitter user's vocabulary can be used to infer their interests as they vary through time, using the TRI word embedding technique.

[85] For example, if there were twenty topics modelled in LDA, where document 1 was associated with topic 3 with a probability of 0.4, topic 4 with a probability of 0.3, topic 19 with a probability of 0.2 and topic 9 with a probability of 0.1, the LDA system would extract the 4 works most closely associated to topic 3, the 3 works most closely associated to topic 4, the 2 works most closely associated to topic 19 and the single word most closely associated to topic 6.

[86] As explained by Röder *et al.,* "the *u_mass coherence* accounts for the ordering among the top words of a topic... Word probabilities are estimated based on document frequencies of the original documents used for learning the topics." [70]

In order to determine the effectiveness of the TRI user profiling system, the generated user vectors for each time bin are compared against the ground truths discussed in *Section 5.1.1.2*. As discussed in *Section 3.2.4.1*, Trec Eval is used to generate a series of metrics which will allow for a quantitative, relevance-based evaluation of the system.

The design of the experiments was heavily influenced by the limited time available within which to complete the evaluation, which had been significantly impacted by the unavailability of the existing ground truth data. Thus, the main criterion for analysis of the solution is based on the variation in performance of the model across different time periods.

5.1.2.1 Temporal Random Indexing

As the primary focus of the implementation, the TRI user profiling system was almost completely prepared for performing an evaluation at this stage.

In order to produce data that can be evaluated against the Word2Vec model and ground truth data, the final required component of the TRI system is a means of determining the words that are most similar to a user's vector for a given time period. This was achieved using a priority queue and the *Cosine Similarity* measure[87], in combination.

- For each time bin, each user embedding vector was compared to the word embedding vectors in the vector dictionary.
- Using the *k-nearest neighbours* algorithm, the top-ten word vector values determined to be most similar to the user vector are recorded in a file, which is formatted in a particular style for compatibility with Trec Eval.

5.1.2.2 Temporal Word2Vec

As motivated in *Section 3.2.4.2*, the baseline implementation chosen for comparison to TRI was a temporal extension of the Skip-gram Word2Vec model. The implementation of this model from the *gensim* Python library was leveraged in the development of the baseline model: Thus, the development work on this component was achieved using Python.

In order to perform an accurate comparison, the same processes applied in the TRI model are also applied to the Word2Vec model, including the data pre-processing and cleaning stages.

[87] Details regarding the Cosine Similarity can be found in both *Section 2.3.2* and *Appendix A.3*.

To capture the temporal aspect of the data, the Word2Vec model was applied to each of the time bins separately. This required the training of a unique NN for each time bin in the data[88,89].

5.1.2.2.1 Training the Word2Vec Model

In order to train a NN model for a time bin, it was first required to first aggregate and cleanse all of the data present in the IMDB, as described in the implementation of the TRI system in *Section 4.1.2*.

When the data has been aggregated and cleaned, it is then passed as input into the *gensim* library implementation of Word2Vec. In order to keep the comparison of Word2Vec and TRI consistent, a sliding context window of size 3 is applied to the text for Word2Vec as it had been for TRI.

The actual generation of the word embeddings using Word2Vec is straight-forward, simply requiring the import of the Skip-gram Word2Vec model from *gensim*, and the provision of the required arguments to train the NN. Each NN was trained for a given time bin.

5.1.2.2.2 User Embedding Generation

When Word2Vec is trained, the next process is to generate the user embedding vectors. This is achieved by accumulating the cleaned data used to train the model, split into its respective tweets for each user.

When this is obtained, the next step is compute a vector for each individual tweet. This is done by querying Word2Vec to obtain the vector embedding value for each word in the tweet, and computing the normalised centroid of the word vectors in the given tweet as shown in *Figure 5.1*.

[88] This approach is very compute-intensive. For example, 421 NNs are generated for the 'week' time bins, and 10 NNs generated for the 'year' time bins of the dataset.

[89] It is assumed that the reader has a basic knowledge of machine learning and NNs, and thus a detailed explanation of the operation of the Word2Vec model is not provided. However, in the event of a lack of knowledge, a tutorial on Word2Vec is provided as means of explanation. [91]

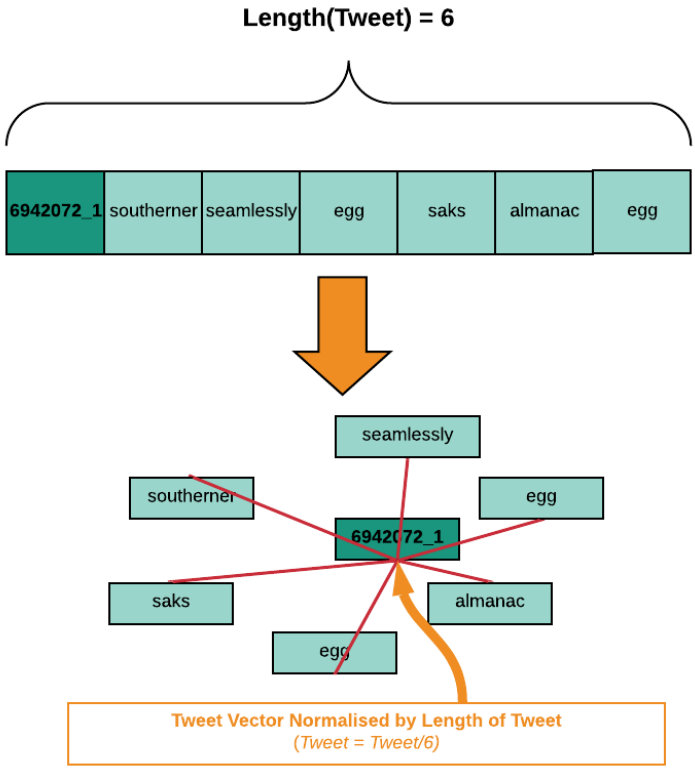

Figure 5-1: Tweet-Word Co-Occurrence to Tweet Vector Example
This image provides An illustration of the generation of a tweet vector in the Word2Vec implementation. It can be seen that the vector for a tweet is computed as the centroid of the word vectors corresponding to the words in the tweet, normalised by the length of the tweet.

When this process is complete, the final embedding vector generated is the user embedding vector, which is computed as the centroid of the tweet vectors in a similar manner to that used in TRI.

5.1.2.2.3 Results From Word2Vec

In order to determine the top-ten most similar word embedding vectors to each user vector, each user vector is passed as an argument to the *most_similar* function defined by the *gensim* library, which computes the Cosine similarity between the specified user vector and the word embedding vectors for that time bin. As with the corresponding results generated by TRI, these top-ten words are recorded in a file formatted in the style required by Trec Eval.

5.1.3 Generating Evaluation Metrics with Trec Eval

As discussed, the Trec Eval tool is used for the purpose of evaluating the TRI user profiling system against both the ground truths and the baseline Word2Vec implementation.

Trec Eval operates by consuming two files: the *qrels* file, and the *results* file.

1. **Qrels File**

   This file contains the set ground truth relevance judgements, each of which corresponds to a particular user. Each row of the file is required to contain a number of fields:

   - *query-id*

     This field is a unique alphanumeric sequence used to identify the subject of the relevance evaluation: In this case, the ID of the user vector[90].
   - *document-id*

     This field corresponds to the *k* most similar words to the given user for the given time bin: Thus, this field is populated with the ground truth values generated by the LDA algorithm.
   - *relevance*

     This field corresponds to the binary relevance judgement[91] for the user vector. Since all ground truth values are considered relevant to the user in the given time bin, the value of this field for row is static with value 1.

2. **Results File**

   The results file contains relevance judgments generated by the system being evaluated, presented as a ranking of documents where each row corresponds to a particular user. The relevant fields in this file are as follows:

   - *query-id*

     As with the *qrels* file, a unique alphanumeric sequence to identify the subject of the relevance evaluation.
   - *document-id*

     As with the *qrels* file, this field corresponds to the *k* most similar words to the given user for the given time bin.
   - *rank*

     This field contains an integer value which represents how similar the given word is to the given user. In this research, the most similar word is assigned the value 0 and the least similar the value 9.
   - *score*

     This field is the similarity score between the user and the word, where the most

[90] The ID for the user vector was composed of the user's unique Twitter ID number and the date corresponding to the start of the time bin for which the vector was generated.
[91] The formula for the binary relevance model is discussed in *Appendix A.3*

similar word will have the highest score. In this research, the score equates to the cosine similarity value obtained for the word with respect to the user.

For each time period considered, the relevant *qrels* and *results* files were provided as inputs to the Trec Eval tool, which generated a set of standard evaluation metrics as a result. The discussion of these results is the subject of *Section 5.2.3*.

5.2 Results

This section provides a series of results and measurements obtained as part of the evaluation of the TRI user profiling system. These include (i) general system metrics regarding the storage memory and computation time required by both TRI and Word2Vec, and most importantly (ii) the performance metrics obtained using Trec Eval.

5.2.1 Memory Requirements

Given that word embedding techniques both use and generate large quantities of data, the amount of storage memory required to perform the necessary processes is an important consideration. The memory requirements for both the TRI and Word2Vec temporal user profiling implementations are presented and discussed in this section. Additional memory statistics are included in *Appendix A.5* for reference.

5.2.1.1 Memory Requirements for Temporal Random Indexing

The amount of storage memory required by the various components of the TRI system is shown in *Table 5.1*. Some interesting observations about this data are as follows:

- It can be seen that as the time period increases in size, the amount of memory required to store the corresponding data decreases.
- It can also be observed that for smaller time periods, the vector representations generated require a significantly larger amount of storage memory when compared to the co-occurrences matrices and the relevance judgements.

Time Period	Co-Occurrence Matrices (MB)	Embedding Vectors (MB)	Relevance Judgements (MB)
Week	593	3072	74.2
Month	535	1617.92	20.3
Quarter-year	461	971	7.53
Semi-year	454	711	3.98
Year	358	474	2.49

Table 5-1: Storage Memory Requirements for TRI
This table shows the amount of storage memory required by TRI for (i) the co-occurrence matrices, (ii) the user and word embedding vectors, and (iii) the relevance judgements generated for each time period.

A number of charts were generated using the above table of data, all of which can be found in *Appendix A.6.1* except for the chart shown in *Figure 5.2*. This chart illustrates the total amount of storage memory required by the TRI system for each time period.

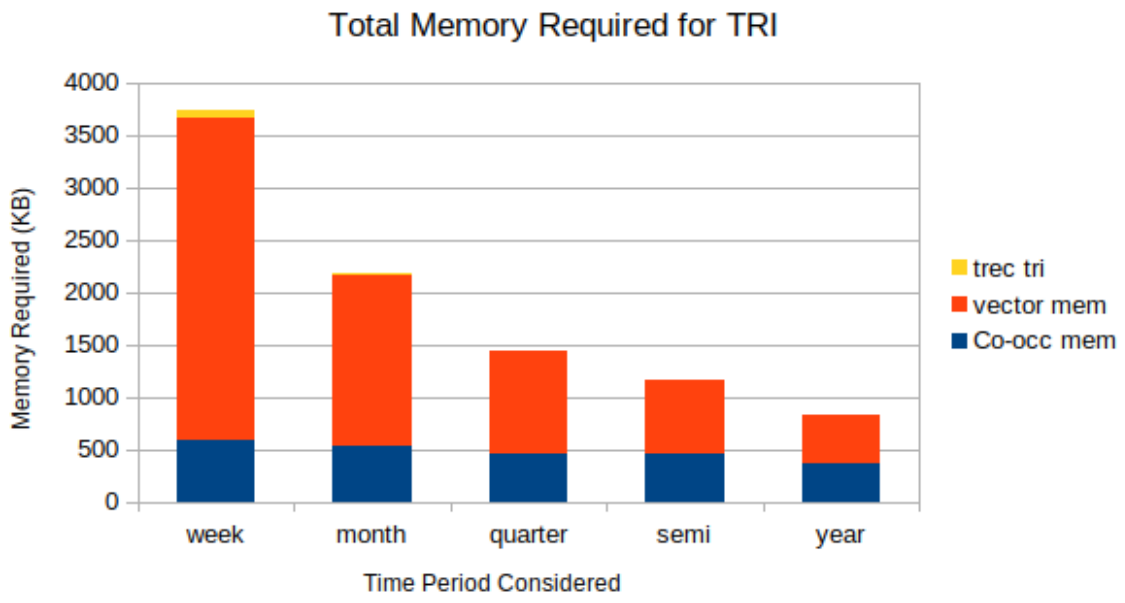

Figure 5-2: Total Memory Requirements for TRI
The above chart illustrates the combined memory requirements for the TRI implementation.

From the plot, it can be clearly observed that for each time period, the co-occurrence files require approximately the same amount of storage memory, and also that the relevance judgements required the least amount of memory to store. As is to be expected, this plot also demonstrates that it is the user and word embedding vectors which are the most demanding in terms of storage memory requirements, with the smaller time periods requiring the most memory for storage.

5.2.1.2 Memory Requirements for Temporal Word2Vec

The amount of storage memory required by the various components of the temporal Word2Vec system is shown in *Table 5.2*. In it, the same trend that was observed for the TRI memory requirements in *Table 5.1* can also be observed: As the time period increases in size, the amount of memory required to store the corresponding data decreases.

Time Period	*NN Models (MB)*	*Relevance Judgements (MB)*
Week	4,440	72.6
Month	2,560	20.3
Quarter-year	1,522	7.58
Semi-year	1,030	3.66
Year	793	2.50

Table 5-2: Storage Memory Requirements for Word2Vec
This table shows the amount of storage memory required by Word2Vec when generating (i) the Word2Vec NN models, and (ii) the relevance judgements, for each time period.

In the same manner as with TRI, a series of bar charts were generated based on the data in *Table 5.2*, all of which can be found in *Appendix A.6.2* except for the plot shown below in *Figure 5.3* which illustrates the total amount of memory required for each time period.

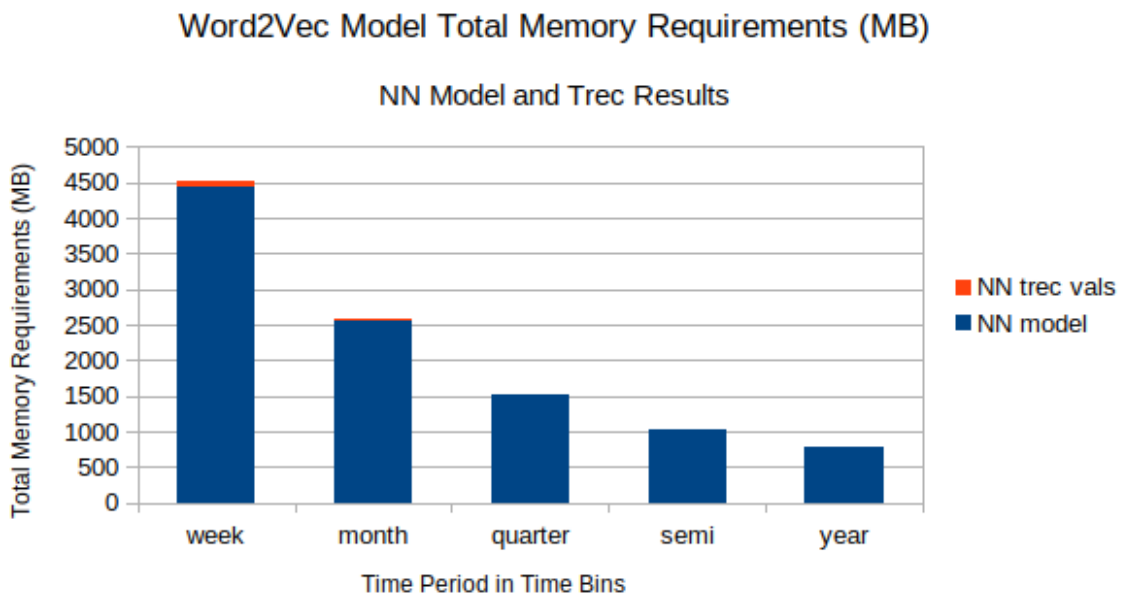

Figure 5-3: Total Memory Requirement for Word2Vec
This graph illustrates the combined memory requirements for the Word2Vec implementation in this research.

From the plot, it can be clearly seen that for each time period, the NN models consume by far the greatest amount of memory, with the relevance judgements used by Trec Eval consuming far less. The same trends in this graph can be seen as those in *Figure 5.1* that is, as the time period increases from week to years, the amount of memory consumed decreases rapidly.

5.2.2 Time Requirements

As part of the system performance evaluation, the time requirements of both the TRI and temporal Word2Vec systems were also measured and are discussed in this section. This is a particularly important criterion for the viable use of TRI operating at scale in a live environment. As before, additional statistics are included in *Appendix A.5* for reference.

5.2.2.1 Time Requirements for Temporal Random Indexing

Table 5.3 displays the amount of time taken by the TRI system to generate the co-occurrence matrices, the embedding vectors and the relevance judgements.

Time Period	**TRI Computation Time (ms)**		
	Co-Occurrence Matrices	***Embedding Vectors***	***Relevance Judgments***
Week	14,605,165	84,260	35,996,713
Month	26,516,115	64,997	19,253,673
Quarter-year	70,548,021	58,820	12,606,443
Semi-year	240,236,094	38,183	9,025,311
Year	298,251,445	31,390	5,677,237

Table 5-3:Time Required to Run TRI
Illustrates the amount of time required by TRI generate the co-occurrences matrices for each time period, and then the time required to generate the vector embeddings; which is significantly less than the co-occurrence generation

Similarly to the analysis of memory requirements in *Section 5.2.1.1*, this table was used to generate a number of bar charts, all of which can be found in *Appendix A.6.3* except for the plot shown below in *Figure 5.4*, which illustrates the total amount of time taken to run the TRI system for each time period.

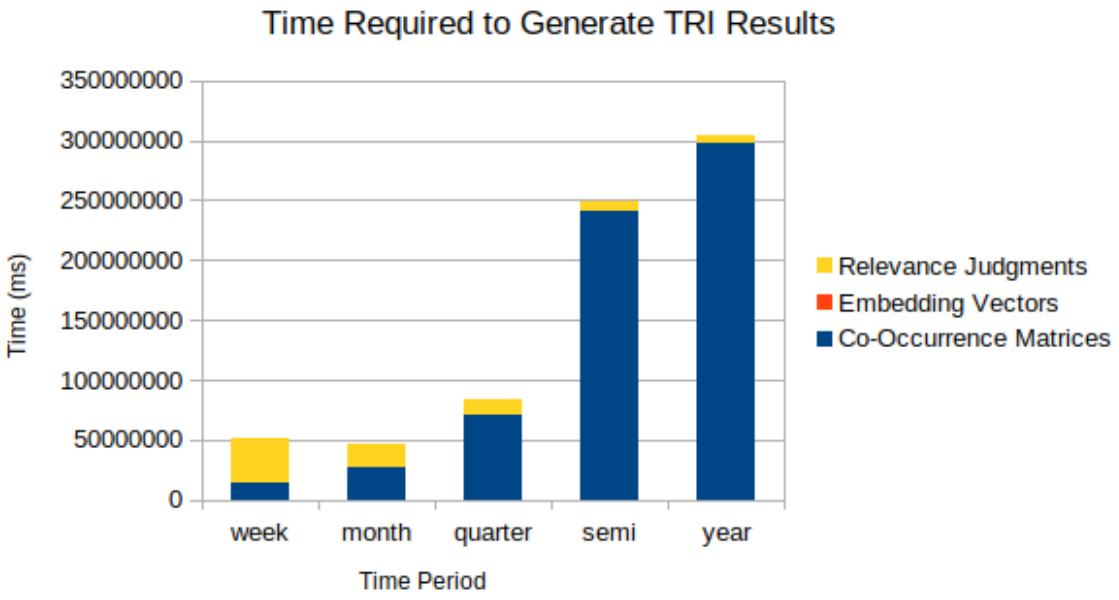

Figure 5-4: Total Computation Time Requirements for TRI
This graph illustrates the total computation time requirements for the TRI implementation.

One striking observation here is the amount of time taken to generate the co-occurrence matrices for the semi-year and year time periods: Both of these took over 2 days to execute in their entirety. The time taken to compute these dwarfs that of the generation of the embedding vectors and relevance judgements to the point where they cannot be observed on the plot.

5.2.2.2 Time Requirements for Temporal Word2Vec

Table 5.4 displays the total amount of time taken by the temporal Word2Vec system to train the NNs and generate the relevance judgements.

Time Period	Word2Vec Training Time (ms)
Week	6,195,928
Month	4,158,764
Quarter-year	3,684,156
Semi-year	3,328,882
Year	3,110,383

Table 5-4: Total Time Requirements for Temporal Word2Vec
This table illustrates the amount of time required by Word2Vec to train the NN models and to obtain the results for Trec Eval for each given time period.

A single bar chart was generated using the data present in *Table 5.4*, which is shown in *Figure 5.5*. A clearly observable trend in this data is the decrease in time taken to complete the training of the Word2Vec model for increasing time period length. This observation can be explained by the number of NNs that are required to be trained for each time period: For example, there are 421 instances of the time period 'week', each of which requires a separate NN to be generated, compared to only 10 instances of the time period 'year'.

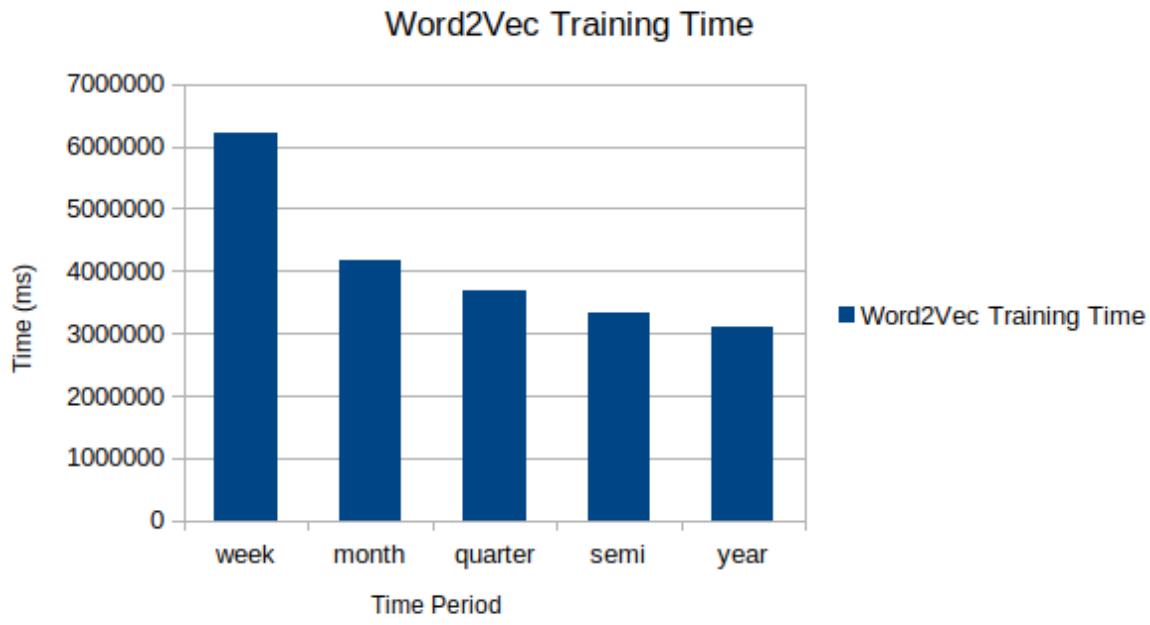

Figure 5-5: Total Time Requirements for Temporal Word2Vec
This graph illustrates the combined memory requirement for the Word2Vec implementation in this research.

5.2.3 Relevance Results

The full set of results that were generated using Trec Eval as described in *Section 5.1.3* are included in *Appendix A.7* of this document.

Shown below in *Table 5.5* and *Table 5.6* are a subset of the results obtained using Trec Eval, corresponding to the Mean Average Precision (**MAP**) and precision at 5 (**P_5**) respectively for both TRI and temporal Word2Vec.

Mean Average Precision		
	TRI	**Temporal Word2Vec**
Week	NA[92]	0.0218
Month	0.0359	0.0082
Quarter-year	0.0077	0.0040
Semi-year	0.0097	0.0006
Year	0.0026	0.0002

Table 6-5: MAP Score for TRI and Temporal Word2Vec
This table provides the MAP results obtained for the TRI and temporal Word2Vec user interest inferences as computed by the Trec Eval library.

Precision @ 5					
	Week	**Month**	**Quarter-year**	**Semi-year**	**Year**
TRI	NA	0.0794	0.0196	0.0243	0.0064
Temporal Word2Vec	0.0328	0.0126	0.0063	0.0011	0.0006

Table 5-6: Precision @ 5 Score for TRI and Temporal Word2Vec
This table provides the P@5 results obtained for the TRI and temporal Word2Vec user interest inferences as computed by the Trec Eval library.

5.3 Discussion

This section provides a detailed discussion of the results outlined in *Section 5.2*. This discussion includes an assessment of the memory and time measurements obtained, but focuses more heavily on the relevance results obtained from Trec Eval. Further to this, a discussion regarding the objectives set out in *Chapter 3* of this document is provided which assesses whether the objectives were met and to what degree.

5.3.1 Discussion of Results and Measurements

[92] It should be noted that there was an issue in generating the results for TRI for the week time period. The issue, which occurred with Trec Eval, could not be rectified due to time constraints, and thus no results were obtained for TRI for the week time period.

This section provides a quantitative evaluation into the implementation of TRI with respect to the Word2Vec implementation developed in this project. This section focuses on three different modes of comparison namely, i) the time requirements for the implementation and, ii) the metric based evaluation of the system.

5.3.1.1 Discussion of Memory Requirements

It is clear from the results and observations of *Section 5.2.1* that the amount of memory required for both the TRI and temporal Word2Vec user profiling systems are on broadly the same scale.

Time Period	*TRI Total Memory (MB)*	*Temporal Word2Vec Total Memory (MB)*	*Percentage Improvement upon temporal Word2Vec*
Week	3739.2	4512.6	17.14%
Month	2173.22	2580.3	15.78%
Quarter-year	1439.53	1529.58	5.89%
Semi-year	1168.98	1033.66	(13.09%)
Year	834.49	795.5	(4.90%)

Table 5-7: Comparison of Storage Memory Requirements
This table shows the total amount of storage memory required by both TRI and temporal Word2Vec for each time period, as well as the percentage improvement of TRI compared to temporal Word2Vec

Based on the data shown in *Table 5.7*, it appears overall that the TRI system requires less total memory than Word2Vec for the same time periods. However, it is also clear that TRI performs better in terms of memory requirements for the shorter time periods, but is outperformed by temporal Word2Vec for longer time periods. This is consistent with the findings of existing research which state the Word2Vec scales well to large datasets. [41] This appears to indicate that TRI has a minor significant advantage over temporal Word2Vec when it comes to memory required for storage, but that this depends on the size of the time period – and hence the dataset - being modelled.

5.3.1.2 Discussion of Time Requirements

As it can be seen from the data provided in *Section 5.2.2*, use of the TRI system requires significant time investment, particularly in the generation of the word-word and tweet-word co-occurrence matrices. The process of generating co-occurrence matrices is inherently time-consuming: However, based on the related literature reviewed in *Chapter 2*, it would be

expected that RI should be more time efficient than other co-occurrence based embedding techniques such as LSA and GloVe due to the implicit dimensionality reduction used by RI[93].

Time Period	*TRI Total Time (ms)*	*Temporal Word2Vec Total Time (ms)*	*Percentage Improvement upon temporal Word2Vec*
Week	50,686,138	6,195,928	(718.06%)
Month	45,834,785	4,158,764	(1002.13%)
Quarter-year	83,213,284	3,684,156	(2158.68%)
Semi-year	249,299,588	3,328,882	(7388.99%)
Year	303,960,072	3,110,383	(9672.43%)

Table 5-8: Comparison of Computation Time Requirements
This table shows the total amount of computation time required by both TRI and temporal Word2Vec for each time period, as well as the percentage improvement of TRI compared to temporal Word2Vec

It is clear from *Table 5.8*, which compares the total time taken and percentage difference between the TRI and temporal Word2Vec models for each time period, that RI would not be a contender if time were a critical success factor for the deployment of one of these temporal user profiling systems.

- The table shows that in every instance, the computation time required by the TRI user profiling system is hundreds and sometimes even thousands of times greater than that required by temporal Word2Vec for the same time period. This could potentially be due to the way in which the system was implemented, since no parallelisation or optimisation measures were attempted in its development. If this is indeed the reason for the alarmingly large computation times for the TRI system when compared to temporal Word2Vec, then it is imperative that the system be refactored to take advantage of parallelisation and optimisation measures before considering its use in a live environment.

- Regardless of the reason for the poor time performance of TRI compared to temporal Word2Vec, it seems obvious that if time was of the utmost importance, temporal Word2Vec should be the preferred option. However, TRI has one significant advantage over Word2Vec in that it can incrementally incorporate updates to the dataset. In

[93] It should be noted that there was no research found which specifically investigated which of these embedding techniques was most time-efficient: However, it is a widely stated fact in the related literature that LSA and GloVe are less time efficient due to their explicit dimensionality reduction.

contrast, if an update was applied to data in a Word2Vec-based model, the model would need to be completely re-generated in order to incorporate the updated data[94].

Given more time, a more in-depth analysis of the time requirements of these temporal word embeddings techniques would prove useful, particularly with regard to how on-the-fly updates are handled.

5.3.1.3 Discussion of Relevance Results

As can be observed from the tables provided in *Section 5.2.3* and *Appendix A.7* that the implementations of both TRI and Word2Vec are not very performant in their inference of users' interests in their current configuration. However, it is clear from the results that TRI outperformed Word2Vec in all performance measures calculated using Trec Eval. The same observation can be made from the line plots shown in *Figures 5.6* and *5.7*, which correspond to the MAP and Precision@5 values obtained for both TRI and temporal Word2Vec for each of the five time periods.

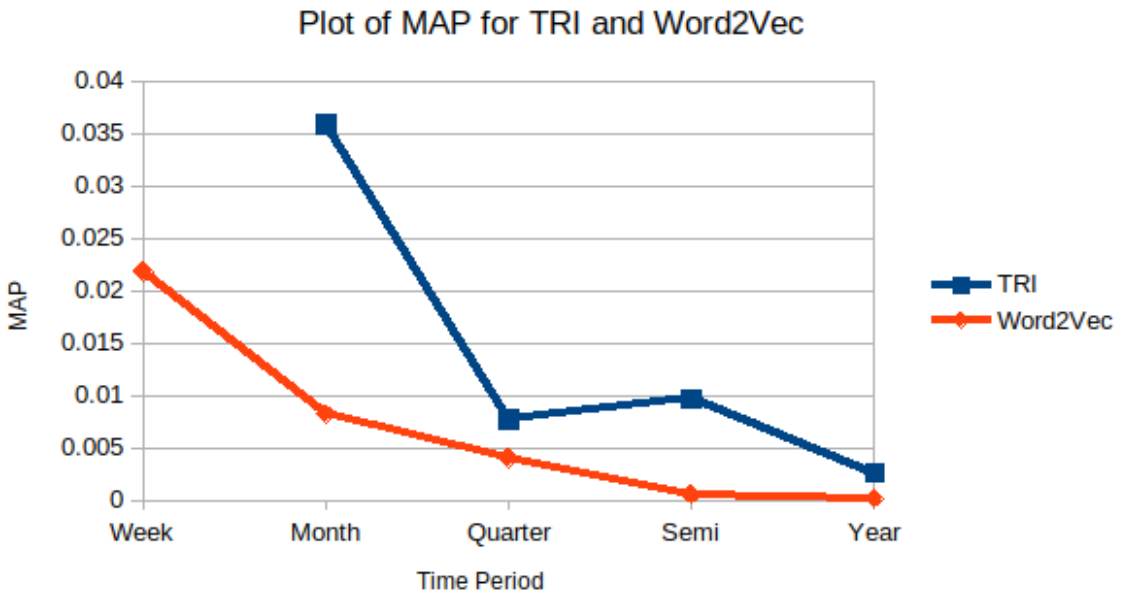

Figure 5-6: Plot of MAP Values for TRI and Temporal Word2Vec
This plot shows the variation of the MAP score for TRI and temporal Word2Vec over each of the five time periods considered in this research

[94] It should be noted that this feature of RI was not tested as part of the evaluation, and so this statement is based on statements made in related literature. [89]

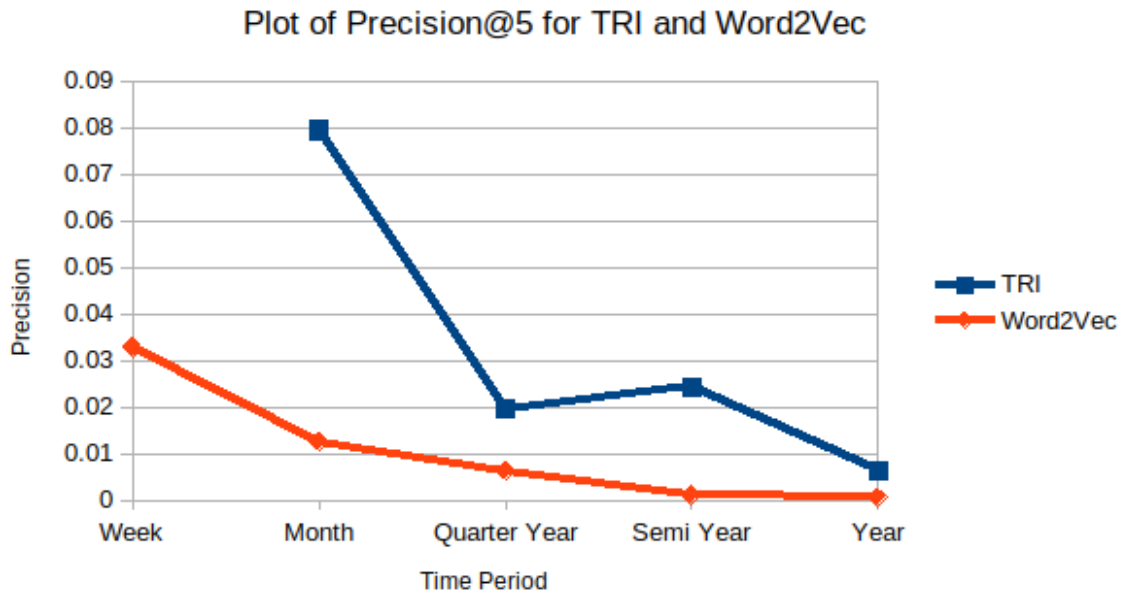

Figure 5-7: Plot of Precision@5 Values for TRI and Temporal Word2Vec
This plot shows the variation of the Precision@5 score for TRI and temporal Word2Vec over each of the five time periods considered in this research

Though it cannot be claimed that either TRI or temporal Word2Vec are providing promising user profiling results, the fact that the results obtained for TRI are better than those for temporal Word2Vec is encouraging. Additionally, the relatively poor performance of Word2Vec in this research does not match the performance claims made by Liang *et al*. in their implementation of a temporal Word2Vec system for the same dataset, [31] indicating that perhaps the variation in results is due to the manner in which the system was implemented in this research. Given that Liang *et al*. also applied additional techniques such as Kalman Filtering to their model, it could be hypothesised that the overall performance of the both models would be improved if the implementation steps applied in this research were more similar in nature to those used by Liang *et al*.

5.3.2 Discussion of Research Objectives

This section provides a discussion of the research objectives originally set out in *Chapter 3* of this document. Each of the research objectives was defined to support in the realisation of the research question, which was defined as follows:

To what extent can a user's vocabulary on an online social network be used to infer their interests as they vary through time using a previously unexplored word embedding technique?

The following subsections address whether the supporting research objectives were achieved, and to what degree.

5.3.2.1 Discussion of Research Objective 1

The first research objective defined in *Chapter 3* of this document is as follows:

To select an appropriate word embedding technique to address the research question. Such a word embedding technique should demonstrate high potential for temporal analysis of OSN user data.

Based on the evaluation performed in this chapter, it is reasonable to consider that this objective was fully achieved.

- TRI was identified as the embedding method of choice for this research, given its substantial demonstrated advantages when compared to alternative techniques as described in *Section 3.2.1.3*.
- Given the fact that TRI was used to successfully implement a temporal user profiling system which produced results that outperformed the current state-of-the-art technique, this word embedding technique has proven to be highly appropriate in addressing the research question.

5.3.2.2 Discussion of Research Objective 2

The second research objective defined in *Chapter 3* of this document is as follows:

To devise an effective approach to representing users by the word embeddings generated from their OSN data.

In the case of this objective, it can be said that it was partially achieved.

- As part of this research, a novel approach to generating a user embedding vector from each user's tweets was devised. This approach attempted to characterise each user by the word embeddings that were determined to best describe their interests, computed as the centroid of their tweet vectors.
- However, the generated embedding vectors performed rather poorly overall when compared against the ground truths. There was not sufficient time to investigate the

cause of this, which could be due to several issues such as not having sufficient context for the users when generating the embeddings.

5.3.2.3 Discussion of Research Objective 3

The third and final research objective defined in *Chapter 3* of this document is as follows:

To define an appropriate evaluation approach which would allow for accurate comparison of the resulting user embedding technique against other similar methods.

In the case of this final objective, it can be said that it was achieved to the maximum possible degree given the circumstances.

- An appropriate evaluation approach was defined and implemented, which involved evaluating the system by comparing the user vectors generated using TRI with those generated using temporal Word2Vec, making use of the Trec Eval tool.
- The degree to which the evaluation could be classified as 'accurate' is a contentious point, due to the issue with obtaining the original ground truth data which resulted in the automated generation of ground truth data using the LSA algorithm.

Considering this, the results obtained from the evaluation would be more meaningful and almost certainly more accurate if the ground truth data had been annotated manually.

5.4 Summary

It is clear that the unpredictable nature of conducting research was a barrier in realising the full potential of this research, insofar as that the results obtained are clearly neither performant nor demonstrate how well embedding methods can perform in the task of temporal user profiling.

However, although the results obtained were poor, TRI was demonstrated to outperform temporal Word2Vec in user inference. This is a significant finding, since temporal Word2Vec is currently considered the state-of-the-art technique in temporal user profiling. Regardless, there is much room for further research to be conducted into improving upon the approaches of this research.

Chapter 6 Conclusions and Future Work

The work carried out to answer the research question set out in *Section 3.1* yielded many interesting insights into the application of TRI to temporal user profiling, particularly regarding how such systems could be most effectively constructed to model users based on their vocabulary. This chapter is a reflection on the research conducted and the insights derived from the findings.

Section 6.1 explores a number of potential avenues for expanding upon the research conducted in this project. These include potential improvements to the proposed approach, as well as opportunities to conduct additional work building on the foundations laid by this research.

Finally, *Section 6.2* draws some conclusions regarding the outcomes of the research, focusing on the original elements of the research objectives.

6.1 Future Work

An end-to-end TRI-based user profiling system was successfully developed in this research. However, despite the promising results obtained, there is much work that could be done to both improve upon them and to explore additional related avenues of research. A number of identified opportunities for future work are discussed in this section.

6.1.1 Datasets and Enrichment Techniques

In this research, the data considered was intentionally limited to text generated from tweets present in the dataset, with some enrichment from scraping the content of live URLs embedded in the tweets. There are many additional avenues of research that could be considered based purely on the data used in this research. The following subsections discuss some possible future work in this regard.

6.1.1.1 Use of Alternative OSN Datasets

This research focused on the use of a Twitter dataset generated by UCL's Big Data Institute to generate temporal user profiles using TRI. Whilst this decision was well-informed, this author believes that there is huge potential in the application of TRI to data from other OSNs: For

instance, much of the user profiling research discussed in *Section 2.1.2.2* has been conducted using OSNs such as Facebook and LinkedIn, in some cases making use of datasets from multiple OSNs in the same research.

The research community could benefit from research into the application of TRI to alternative OSN data, or even the use of multiple OSN datasets which could provide additional data enrichment. There are however some important considerations to be accounted for before making such a decision: As discussed in *Section 3.2.1.1* of this document, considerations such as the simplicity of data structure, open availability of the data and suitability of the data for the task at hand are all crucial. In the age of GDPR, it is also prudent to consider the ethical use of such data and any complications which could arise from its use.

6.1.1.2 Use of an Alternative Twitter Dataset

One of the primary limitations of the evaluation of TRI in this research is based on the nature of the Twitter dataset which was used. Given that almost 4 years had passed between the construction of the UCL Twitter dataset and this research, much of the content had expired: This was most clearly seen in the enrichment of the tweets using URL content, where only 5.7% of the URLs extracted from tweets were found to still be active. Though this is partially a symptom of the fact that the data collected is inherently variant with time and thus becomes less relevant with age, the use of a more recently collected dataset would be likely to yield better results.

Another interesting opportunity in this regard would be to accumulate a dataset over a longer period of time, which would give rise to more possibilities in terms of time periods considered and thus result in greater statistical accuracy based on a greater number of results. However, one would assume that as the time period considered grows larger, the dataset would tend toward being static and result in reduced visibility of patterns of user interests such as cycles and spikes[95]. An interesting tangential observation is the temporal nature of the relevance of data itself, which could also be an interesting avenue of future research to consider.

[95] For example, if one were to take the time period of years 2016-2036 and model the interests of a set of users by a time period of decades, it is possible that the model would not capture the significance of data such as if a user tweeted extensively about Brexit in 2019 when they were due to exit the EU. This hypothetical explanation simply illustrates that it may be possible that larger time periods could result in reduced insights being gained about users' interests.

6.1.1.3 Alternative Enrichment Techniques

With regards to enrichment of the dataset used, there are several opportunities for extending the research conducted here. One very obvious opportunity which was referenced in *Section 3.3.1.2* is the analysis of emojis to provide additional context: Investigating a user's sentiment for a given set of words and thus interests could substantially augment the research done here using inferences regarding their feelings about their interests[96]. Similar enrichment could be conducted using hashtags; However, there are some additional challenges that can occur with hashtags such as spelling mistakes and word concatenation which have the potential to make sentiment analysis using hashtags more difficult.

Another enrichment approach identified during the implementation phase of this research was the use of the Wayback Machine API to scrape URL content from archived versions of inactive URLs: However, time did not permit for this to be investigated further. Some potential issues that may hinder this approach could extend to data protection issues, where URLs which were once public and are now private could be included in the inactive URLs, and thus such an approach would require further research into data regulations.

Finally, future work could build upon the foundations laid in this research by investigating embedding video and other graphic content in tweets and using ML content detection techniques to describe the contents, which could also be used for enrichment purposes. This author believes that to build such an extension to the system developed in this research would require substantial preliminary research and a thorough design in order to be successful.

6.1.2 Embedding Techniques

This research was conducted based on the identified potential of word embedding techniques for user profiling; in particular, relating to the use of temporal word embedding techniques for temporal user profiling. Though RI was selected as the technique of choice for this research, there is substantial work remaining to determine the general effectiveness of word embedding techniques for both temporal analysis and user profiling.

[96] Using a hypothetical example, if a user tweets about current U.S. President Donald Trump and then tweets about Democrat Bernie Sanders, it might be possible to infer from their use of emojis whether the user is positively interested in Bernie Sanders and negatively interested in Donald Trump, or the opposite.

As described in *Section 2.3.3* of this document, there exists a wealth of word embedding techniques.

- For most of these, there has been no research found which investigates their potential in user profiling. State-of-the-art techniques such as FastText and GloVe are excellent candidates for conducting research into the use of alternative word embedding techniques for user profiling: For instance, there is a strong case for the use of GloVe in comparing the evolution of different users over time given its excellent performance in similarity tasks. [47]
- Additionally, there has been very limited research found regarding temporal word embedding techniques in general as is seen in *Section 2.3.3.3* of this document. The extension of state-of-the-art techniques such as those listed in *Section 2.3.3.2* to consider temporal variations in language use would be an excellent contribution to the research community.

Another avenue which could be investigated is the inference of temporal variation in user expertise through the use of temporal word embeddings. The idea of this is that expertise could be represented as an embedding vector, where instead of representing a word the embedding would represent job positions or expertise. Such work could contribute significant enhancements to the expertise modelling research described in *Section 2.1.2.2.2*: In such research, datasets from alternative or additional OSNs such as LinkedIn are likely to prove useful.

6.1.3 Improvements to System Design and Implementation

Based on the discussion provided in *Section 5.3.2*, it is clear that the user profiling system could be designed to be substantially more scalable, particularly in relation to computation time. The biggest performance bottleneck was observed for the generation of the co-occurrence matrices: Given that over 7.77 million unique words were identified by TRI for the year time bin of 2014, and that the dataset contained tweets for every year in the period 2006-2015, generation of the full set of co-occurrence matrices for the year time bins took over 3 full days to be processed by the system. Though performance and scalability were not the focus of this research, such performance is clearly not sustainable for web-scale applications.

Improving the architecture of the system to potentially achieve web-scalability would require placing emphasis upon performance optimisation in the design of the system, such as the use of parallelisation where possible.

- The use of design patterns such as the work stealing pattern, where a set of threads are spawned and execute a repetitive task in a parallel fashion, would almost certainly have improved the performance of the system. Based on a brief analysis, this multithreaded pattern could be implemented in the system in two ways:

    a. **Single Shared Co-Occurrence Data Structure:**

    In this approach, the parallelisation occurs on a row-by-row basis. Each row of the co-occurrence matrix would be generated by a separate thread, and used to update a shared co-occurrence data structure.

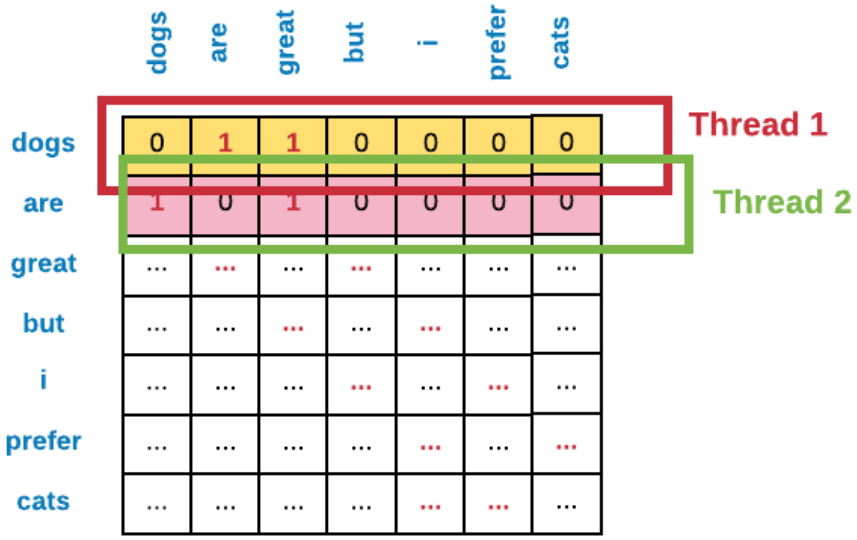

Figure 6-1: Proposed Parallelised Solution 1
This provides An illustration of the parallelised generation of a word-word co-occurrence matrix using a single shared data structure.

    b. **Thread-Specific Co-Occurrence Data Structures**

    This is best described as a divide-and-conquer approach: Rather than multi-threading the generation of the rows of a shared co-occurrence matrix data structure, each available thread takes a portion of the rows to be generated and

updates its own co-occurrence data structure. Once the thread has finished, it can then merge its results with those of other threads[97].

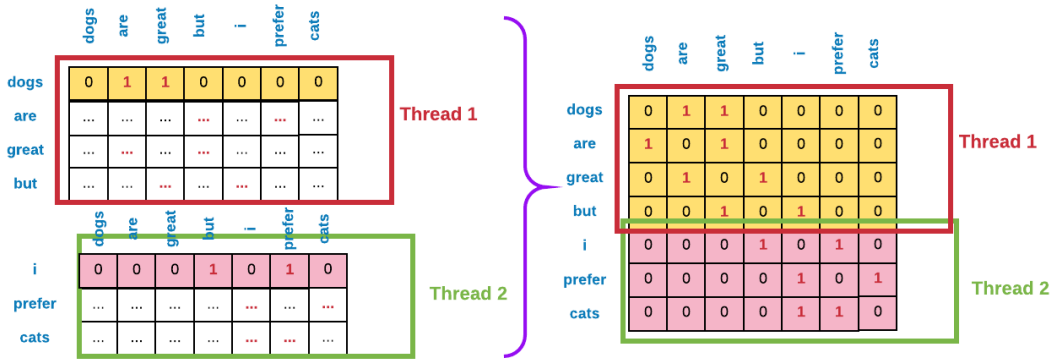

Figure 6-2: Proposed Parallelised Solution 2
This image provides An illustration of the parallelised generation of a word-word co-occurrence matrix using thread-specific data structures.

- Additionally, the use of more optimal data structures and logical conditions would almost certainly have improved the performance of the system.

Due to the computationally intensive nature of the operations being performed in the generation of temporal random embeddings, the overall process is inherently time-consuming regardless of optimisations. However, it should be possible to substantially improve the performance of the system by considering the above enhancements.

6.1.4 Additional Enhancements and Improvements

A number of additional, smaller-scale enhancements and improvements to the TRI user profiling system were identified as follows:

1. **Use of a Dictionary to Reduce Mislabelled Misspellings**

   One idea which was explored but not implemented as part of the system was the use of a dictionary API to reduce the number of words that were mislabelled as being misspellings. Though it would appear that the use of any such APIs would require some financial investment, this would be an excellent means of improving the overall quality of user profiles generated by the system. An alternative to this would be the

[97] In this approach, there would be some additional challenges to overcome in the merging of word-word co-occurrences which would not be encountered in the generation of tweet-word co-occurrences. Thus, some additional work would be required to implement this fully.

development of such an API for the purpose of research, which could be open-sourced and made available to other researchers.

2. **Analysis of Multiple Languages**

   It was not within the scope of this research to consider generating temporal user profiles based on embeddings generated for multiple languages. However, with 29 other languages present in the dataset used in this research alone, there is a clear need to explore the use of word embeddings for temporal user profiling in languages besides English.

3. **Tuning of User Profiling System Parameters**

   There are a number of different parameters of the TRI user profiling system for which an initial value was selected and not varied thereafter. These include crucial parameters such as the context window size in the generation of the word-word co-occurrence matrices, the truncation limit for the time bin dictionaries, etc. The effects of varying these parameters are therefore entirely unknown, and could impact significantly on the performance of the system if investigated.

4. **Comparison of TRI against another Temporal Word Embedding Technique**

   The comparison of TRI to a single state-of-the-art embedding technique was largely due to the time constraints involved in this research. Comparison of the TRI user profiling system against other state-of-the-art techniques such as GloVe, FastText or LSA would almost certainly provide richer insights into the performance of TRI by allowing for a greater degree of comparison.

5. **Alternative Approaches to Generating Ground Truth Data**

   One of the biggest challenges faced during the research was the realisation that the ground truth data for evaluating the system would need to be generated, rather than being able to avail of existing ground truth data. The choice of the LDA algorithm for generating the ground truth data was motivated primarily by time constraints, and so alternative methods of generating annotated ground truth data would likely improve the results obtained from the TRI user profiling system.

6.2 Conclusions and Final Remarks

Although it is clear that this research would have benefited significantly from additional time where (i) more advanced features could have been implemented and (ii) a more thorough evaluation could have been conducted, there was a significant amount of knowledge gained from the exploration and investigation into the state-of-the-art in the domains of NLP and user profiling. The achievements and final conclusions drawn from this research are provided below.

6.2.1 Temporal Random Indexing for User Profiling

One of the significant findings of this research is that TRI is a technique which has great potential in the domain of user profiling. Although the results obtained for the TRI user profiling system implemented in this research project were far from ideal, TRI was found to outperform temporal Word2Vec - a technique which is considered the state-of-the-art in this domain - for all experiments conducted.

It is the opinion of the author that with additional refinement and experimentation, a temporal user profiling system employing TRI could be realised which is performant and scalable enough to be considered for use in live web-scale environments.

6.2.2 The Future of Word Embeddings in Temporal User Profiling

Based on the research conducted, it is clear that the opportunities for word embedding techniques in temporal user profiling applications are immense. As a domain which has seen little research to-date, everything is a possibility - and given the ever-increasing volume of text content generated by users of some of the world's biggest OSN platforms, the growing importance of strong NLP techniques such as temporal word embeddings cannot be overstated.

APPENDIX

A.1 Abbreviations

Abbreviation	Expanded Term
AI	Artificial Intelligence
API	Application Programming Interface
CBoW	Continuous Bag of Words
CV	Context Vector
DH	Distributional Hypothesis
FAIR	Facebook AI Research
GDPR	General Data Protection Regulation
GloVe	GLObal VEctor
IDF	Inverse Document Frequency
IMDB	In Memory DataBase
IR	Information Retrieval
IV	Index Vector
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
MAP	Mean Average Precision
ML	Machine learning
MWE	Morphological Word Embedding

NLP	Natural Language Processing
NN	Neural Network
OCEAN	**O**pen, **C**onscientious, **E**xtraversion, **A**greeable, **N**euroticism
OHE	One-Hot Embedding
OSN	Online Social Network
RGT	Relevance-oriented Ground Truth
RI	Random Indexing
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF	Term Frequency
TRI	Temporal Random Indexing
VSM	Vector Space Model
WSD	Word Sense Disambiguation

A.2 Dataset Statistics

This section provides metadata and statistics relevant to the Twitter dataset collated by UCL's Big Data Institute[98]. [68]

Property	**Value**
Number of Users in Raw Dataset	1,198
Number of Users in MongoDB	1'198
Total Number of Tweets in Raw Dataset	3'658'673
Total Number of Tweets in MongoDB	3'039'118
Total Number of Non-English Tweets	619'557
Number of Normal Tweets in MongoDB	2'447'268
Number of Retweets in MongoDB	591'848
Total Number of Invalid Tweets[99] in Raw Dataset	2
Average Tweet Length in MongoDB	55 characters
Max Tweet Length in MongoDB	290 characters
Min Tweet Length in MongoDB	13 characters
Max Usable Tweets Posted by a Single User	3,243
Min Usable Tweets Posted by a Single User	0
Average Number of Tweets per User	2,042
Number Active URLs Accessed	48,420

[98] The languages illustrated in the table field 'Languages in Dataset' were determined using ISO-639-2 standards.
[99] An invalid tweet is classified as one which either (i) consists of 1 character or less, or (ii) contains no alphanumeric characters.

Number Inactive URLs Accessed	800,140
Languages Present in Dataset	Arabic (AR), Bosnian (BS), Chinese (ZH), Danish (DA), Deutsch (DE), Dutch/Flemish (NL), English (EN), Finnish (FI), French (FR), Greek (EL), Haitian (HT), Hungarian (HU), Indonesian (IN), Italian (IT), Japanese (JA), Korean (KO), Polish (PL), Portuguese (PT), Russian (RU), Slovak (SK), Slovenian (SL), Spanish (ES), Swedish (SV), Tagalog (TL), Tamil (TA), Thai (TH), Turkish (TR), Undefined (UND), Urdu (UR), Welsh (CY)

A.3 Formulae and Metrics

The content in this section describes the two primary ways in which relevance is computed in systems such as those described in this dissertation, along with formulae for the most commonly computed evaluation metrics.

A.3.1 Relevance Formulae

1. **Binary Relevance Model**[100]

   $boolean(A, B) = \begin{cases} 1 \; if \; A=B \\ 0 \; if \; A!=B \end{cases}$; where A is the inferred value or result from a model, B is the ground truth value for the given scenario. If the result equals the ground truth, the value 1 is returned; meaning it is relevance, otherwise 0 is returned; making the result irrelevant.

2. **Cosine Similarity**

   $$cos(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

   Where A is the inferred value or result from a model, B is the ground truth value for the given scenario. The cosine similarity between two values is obtained by the above equation. If the value is 1, then the result and the GT are the exact same, and differing to the binary relevance, the returned value here can be any real number between 1 and -1, where 1 means they're the same, 0 means there's no correlation between the two values and -1 means the values are antonymous to each other; i.e. they have opposite meanings.

A.3.2 Metric Formulae

- **Precision at k (Pre@k)**

  $$pre@k = \frac{TP}{TP + FP} = p(y = 1|\hat{y} = 1) = \frac{|A \cap B|}{|B|}$$

[100] This model is also commonly known as the Boolean relevance model.

where k = max result limit, for the precision of a set of results, there are many different representations that convey the same information, several of which are illustrated here. In simple terms, the precision is the measure of closeness between two sets - i.e. how many correct values are in the results.

- **Mean Average Precision (MAP):**

$$MAP(A,B) = \frac{\sum_{i=1}^{n} AP(A,B)}{n}$$

where AP is the average precision for the set of values of A, B - The results and the GTs. In simple terms, the MAP is the average of the average precisions between the results and GTs sets.

A.4 In-Depth Explanation of Random Indexing Techniques

This section provides in-depth details of the stages involved in both the RI and TRI word embedding techniques for reference.

A.4.1 Random Indexing

The objective of the RI method is to produce an approximation of a co-occurrence matrix similar to that generated in other embedding methods such as LSA. In his paper, [39] Sahlgren describes the co-occurrence matrix *F* as follows[101]:

"*... Co-occurrence matrix* ***F****, such that each row* $\boldsymbol{F}_{\mathbf{w}}$ *represents a unique word* ***w*** *and each column* $\boldsymbol{F}_{\mathbf{c}}$ *represents a context* ***c****, typically a multi-word segment such as a document, or another word... The cells* $\boldsymbol{F}_{\mathbf{wc}}$ *of the co-occurrence matrix record the frequency of co-occurrence of word* ***w*** *and document or word* ***c****. As an example, if we use document-based co-occurrences, and observe a given word three times in a given document in the data, we enter 3 in the corresponding cell in the co-occurrence matrix.*"

In RI, the co-occurrence matrix generated is an approximation of *F*, given by *F'*. The generation of this approximate co-occurrence matrix *F'* can be split into two distinct processes: Namely *index vector* (**IV**) generation and the CV generation.

1. **Index Vector Generation**

   Instead of producing a vector representation of words with very large dimensions as is the approach of alternative techniques, RI makes use of a *maximum dimension space* value, which limits the number of dimensions to a particular value d[102]. Each context i.e. document or word in the corpus, is assigned a unique and randomly generated vector representation called an index vector. The generated IVs are real-valued, sparsely populated and ternary, [39] with the vast majority of values being 0 and a small number of randomly distributed values being either -1 or +1 and can be defined numerically using *Equation A-4-2*.

[101] This description of the co-occurrence matrix *F* matches the description of the co-occurrence matrix *A* generated in LSA.
[102] The value of *d* is usually on the order of thousands. [39]

$$S = \{x \mid x \in -1, 0, 1\}.$$

Equation A-4-1: Ternary Set In Set Notation
This expression defines a ternary vector S as being composed of values x, where x can take on three possible values: -1, 0 or 1.

2. **Context Vector Generation**

Once all of the IVs have been generated, the next step is to generate the context vector. The context vector is a nearly-orthogonal *d*-dimensional representation of a word, corresponding to a row of the approximated co-occurrence matrix *F'*. To generate the CV, the corpus is scanned word-by-word. Each time a word *w* occurs within a given context c^{103}, that context's *N*-dimensional IV is added to the CV for the given word. This results in the representation of the given word *w* as *d*-dimensional CVs that are effectively the sum of the words' contexts. An illustration of the process of generating a context vector for a word or document can be seen in *Figure A-4-3*.

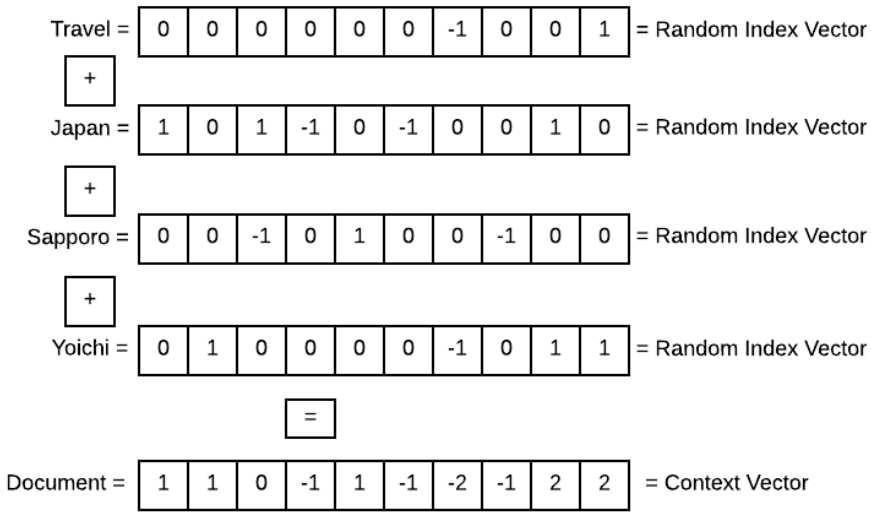

Figure A-4-3: Context Vector Generation
An illustration of how, given a series of words used to define another word, RI is performed. Initially, each word in the corpus is assigned a random IV, e.g. 'travel' is defined by the random IV [0, 0, 0, 0, 0, 0, -1, 0, 0, 1]. Each time a word occurs within the context of the current word, its IV is summed with those of other words that are also within the context.

The result of the RI operation is an approximation of the co-occurrence matrix *F'* of order $k \times d$, whose rows correspond to the *nearly orthogonal* context vectors generated using the previously outlined process. This relationship is described by *Equation A-4-3*.

[103] This context is defined as either a sliding window, or the entire document.

$$F'_{k\,x\,d} \approx F_{k\,x\,n} \mid d \ll n$$

Equation A-4-2: Relationship Between Co-Occurrence Matrix and Approximated Co-Occurrence Matrix
This Illustrates The refined mathematical relationship between the standard co-occurrence matrix and the approximate co-occurrence matrix, which states that F' is an approximation of F such that d is significantly less than n.

The matrix *F'* is an approximation to *F* in the sense that the corresponding rows of both matrices would be either similar or dissimilar to the same degree, but with the benefit of dimensionality *d* being significantly less than *n*. This property makes it possible for RI to achieve the same dimensionality reduction effects as the SVD algorithm in the LSA method, but without requiring a separate dimensionality reduction step to do so.

A.4.2 Temporal Random Indexing

Temporal Random Indexing is a temporal extension of the Random Indexing technique, enabling analysis and investigation into the evolution of word meanings over time.

The first step in the static RI process, whereby a random vector is generated and assigned to each unique word in the vocabulary *V*, is employed in TRI in the same way as in static RI. An extension which incorporates temporality into the approach is added to the word space generation step.

The steps in this new augmented RI process are as follows:

1. **IV Generation**

   From a given corpus *C*, the vocabulary *V* of the *k* words is extracted. A random IV is generated for each word *w* in *V*, according to *Equation A-4-4.*

$$IV_i = randVect(w_i), \text{where } w_i \in V$$

Equation A-4-3: Index Vector Generation
An expression of the formula used to calculate the ith index vector IV_i for the ith word w_i.

2. **Context Vector Generation**

   Next is context vector generation, where CV_i which is computed for each word as the sum of all IVs assigned to the words that occur in the same context as the word w_i ;i.e. $w_i \in c$. This is computed using the equation given in *Equation A-4-5*.

$$CV_i = \sum_{d=0}^{D} \sum_{j=-c, j \neq i}^{c} IV_c$$

Equation A-4-4: Context Vector Generation in Function Notation
This illustrates the formula used to calculate the ith index vector IV_i for the ith word w_i.

The word vector generation equation above is altered to incorporate temporal aspects of words. In order to do this, *C* must first be annotated with timestamps to capture the meaning of words at a given time for a given time period[104]. For each time period, a separate word space is produced. For this, a range of *time bins* must be defined. Let T_k^n denote the *k* number of time bins for a time period *n*. In T_k^n, we define a range from $t_{k-start}^n$ to t_{k-end}^n where $t_{k-start}^n < t_{k-end}^n$. Hence, the temporally augmented word vector equation is given as: $wv_{i,T_k^n} = \sum_{d=0}^{D} \sum_{j=-c, j \neq i}^{c} IV_c$ which is the word space generated for time bin *k*. Hence, there will be *k* word spaces generated for each time period defined by *n*.

From a corpus of text, a *word-word co-occurrence matrix* is generated, an example of which is illustrated below in *Figure A.-4-4*.

	best	lazy	cute	funny	weird	odd
dog	4	0	7	0	0	0
cat	0	15	6	2	0	0
goat	0	0	0	0	12	6

Figure A-4-4: Example Co-Occurrence Matrix
This illustrate a word-word co-occurrence matrix, populated with dummy data. The diagram shows that within a given context, terms such as 'best' and 'cute' occur frequently with 'dogs', whilst 'weird' and 'odd' occur frequently with the term 'goat'. The step preceding this would be to calculate the CV by summing the IVs for each word by their respective multiplicative factor; i.e. $CV_{dogs} = IV_{best}*4 + IV_{cute}*7$ etc

A.5 General System Requirements

This section provides detailed data regarding the storage and computation time requirements of the user profiling system described in this document.

A.5.1 Storage Memory Requirements

[104] Such a time period could be a day, week, month, year, etc. This is not the same as a time *bin*, which is an instance of a time period: For instance, given a time period of 1 year, an instance of a time bin for this time period would be the year 2018.

The storage requirements for the dataset, including the raw data, pre-processed data and HTML content data, are displayed in *Table A-5-1*.

Data	Memory Requirement (MB)
Raw Dataset	10,700
MongoDB Pre-processed Dataset	490
MongoDB HTML Enrichment	25

Table A-5-1: Memory Required For Dataset

The storage requirements of the input data to the system. The 'Raw Dataset' row corresponds to the the publicly available data obtained from S. Liang's BitBucket account, which was used in several of Liang's research papers. [30] [31] The MongoDB rows illustrate the amount of data that was present in MongoDB after the pre-processing described in Section 3.2.1: The dramatic difference between the storage requirements for this and the raw dataset illustrates the amount of redundant data that was present in the raw dataset. The 'HTML Data' row illustrates the amount of data scraped from the embedded URLs.

The storage requirements for the LDA-generated ground truths are displayed in *Table A-5-2*.

Time Bin	File Size (MB)
Week	44
Month	8.7
Quarter-year	3.4
Semi-year	1.68
Year	1.1

Table A-5-2: Memory Required For Ground Truths

A.5.2 Computation Time

The computation times required for the TRI user profiling system to generate (i) co-occurrence matrices, (ii) user vectors and (iii) relevance judgments, for each of the time periods considered, are shown in *Table A-5-3*.

Time Period	TRI Computation Time (ms)		
	Co-Occurrence Matrices	***Embedding Vectors***	***Relevance Judgments***
Week	14,605,165	84,260	35,996,713
Month	26,516,115	64,997	19,253,673
Quarter-year	70,548,021	58,820	12,606,443
Semi-year	240,236,094	38,183	9,025,311
Year	298,251,445	31,390	5,677,237

Table A-5-3: Time Required For Temporal Random Indexing

The time required by the TRI system to, for each time period, (i) generate the co-occurrences matrices, (ii) generate the embedding vectors (which takes significantly less time than the co-occurrence generation), and (iii) generate the relevance judgements.

The computation times required to train the Word2Vec model and generate relevance judgments for each of the time periods are shown in *Table A-5-4*.

Time Period	Word2Vec Training Time (ms)
Week	6,195,928
Month	4,158,764
Quarter-year	3,684,156
Semi-year	3,328,882
Year	3,110,383

Table A-5-4:Word2Vec Training Times

This Table Illustrates The times required by the Word2Vec implementation, for each time period, to train the NN model and to generate relevance judgements.

A.6 Graphs Obtained From Evaluation Data

This section contains graphs generated as part of the evaluation of the research.

A.6.1 Temporal Random Indexing Storage Memory Requirements

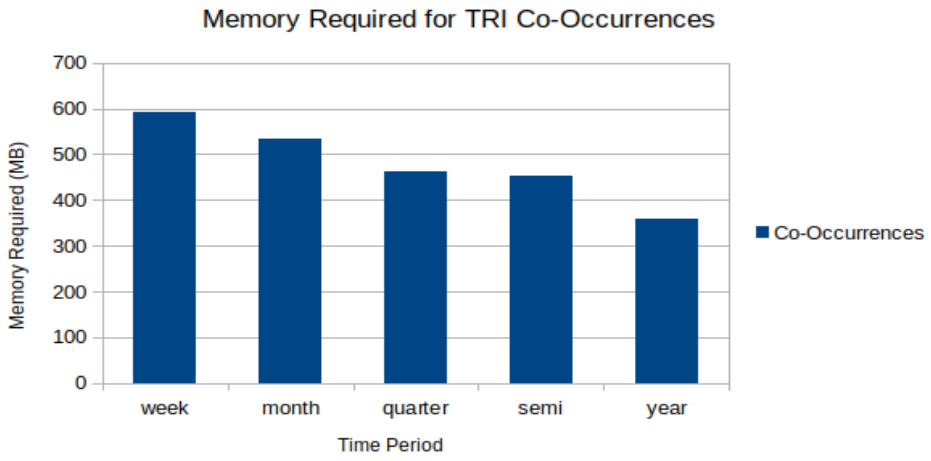

Figure A-6-5: Storage Memory Required for TRI Co-Occurrences
The above graph is a plot of the data provided in the 'Co-Occurrence Matrices' column of Table A.5.3, i.e. the storage memory requirements for the generation of the co-occurrences matrices in TRI.

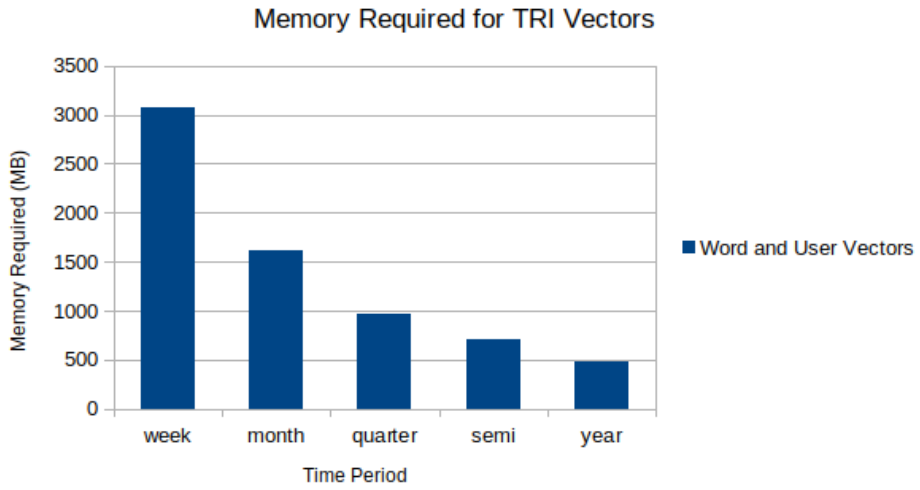

Figure A-6-6: Storage Memory Required for TRI Vectors
The above graph is a plot of the data provided in the 'Embedding Vectors' column of Table A.5.3, i.e. the storage memory requirements for the generation of the user and word vectors in TRI.

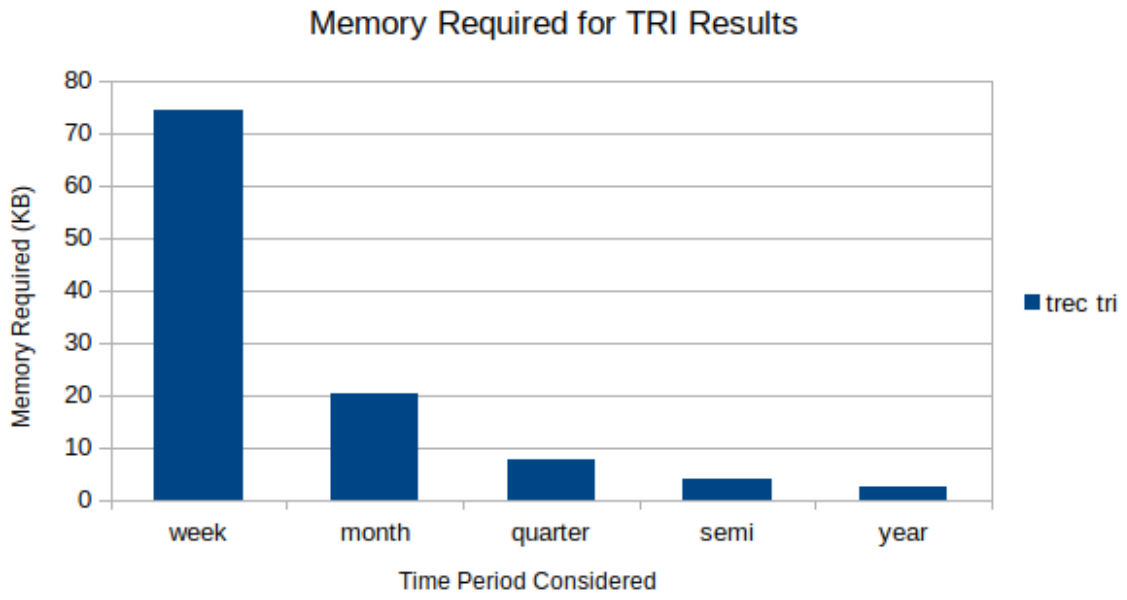

Figure A-6-7: Storage Memory Required for TRI User Interest Inferences
The above graph illustrates the graphical representation of the data illustrated in the User Inferences column of Table 5.2, i.e. the memory requirement for the generation of the user interest inferences in TRI

A.6.2 Temporal Random Indexing Computation Times

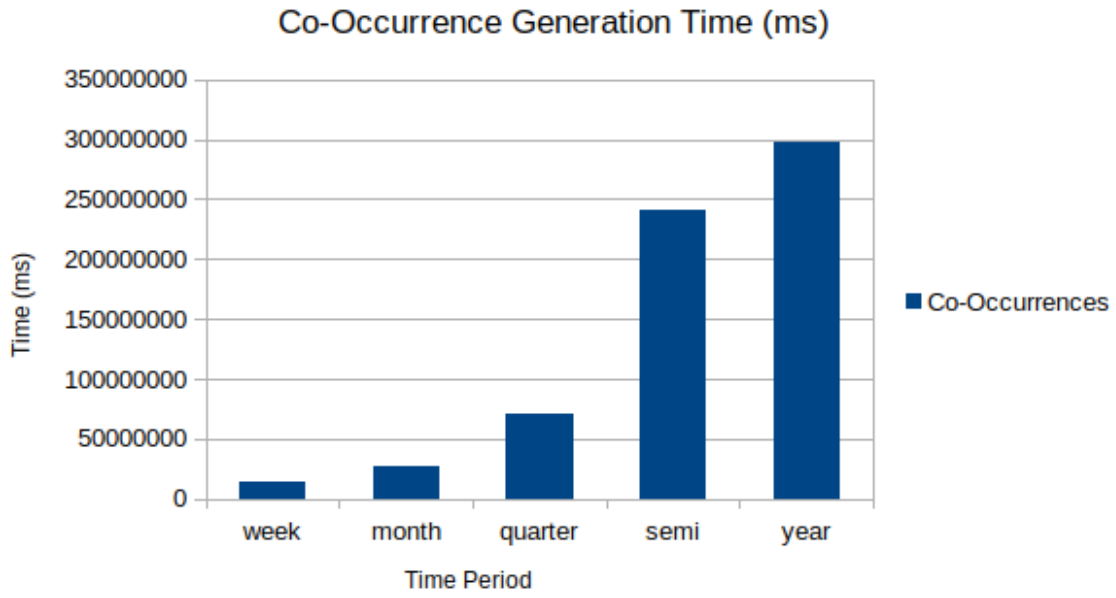

Figure A-6-8: Time Required for TRI Co-Occurrence Generation
The above graph is a plot of the data in the 'Co-Occurrence Matrices' column of Table A.5.3, i.e. the time required to generate the word-word and tweet-word co-occurrences using TRI.

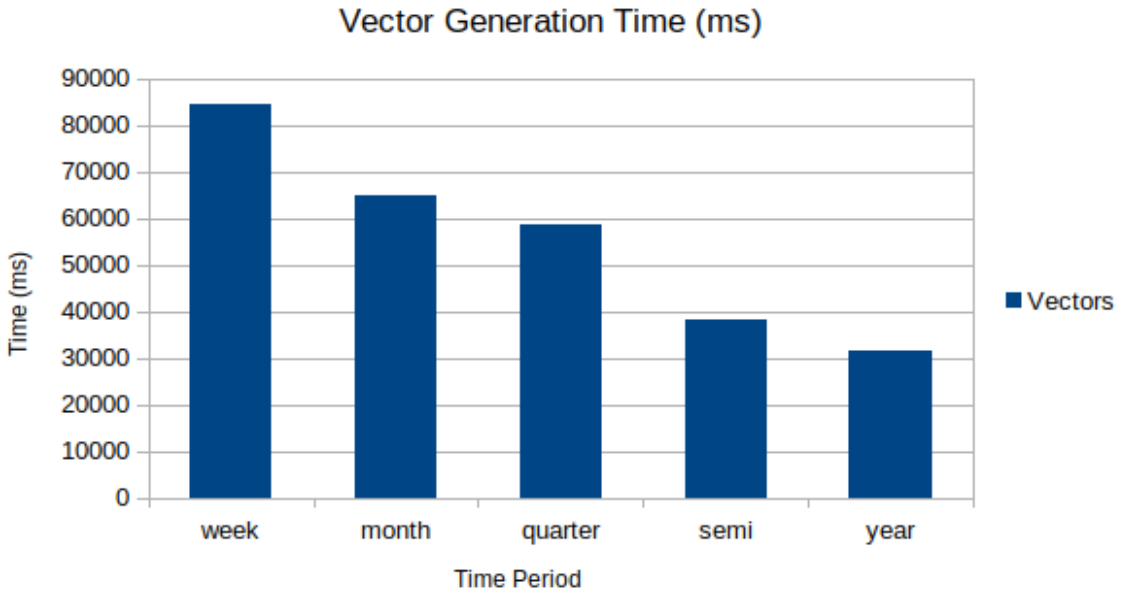

Figure A-6-9 : Time Required for the TRI Vector Generation
The above graph is a plot of the data in the 'Embedding Vectors' column of Table A.5.3, i.e. the time required to generate the user and word vectors using TRI.

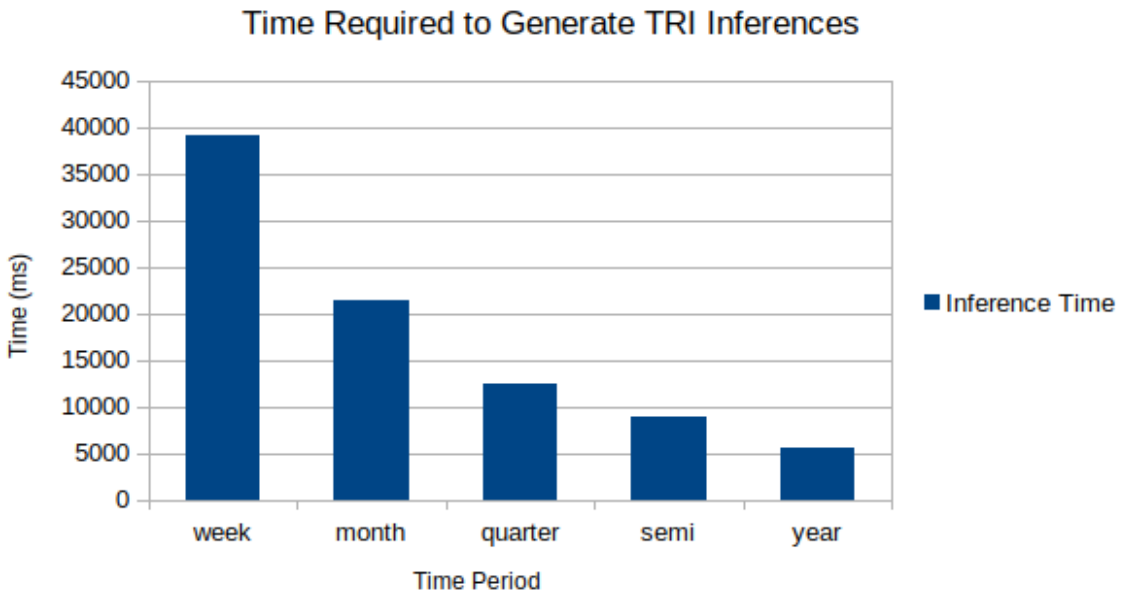

Figure A-6-10: Time Required for the TRI User Inferences
The above graph illustrates the graphical representation of the data illustrated in the Results column of Table 5.4, i.e. the time required to generate the user interest inferences using TRI.

A.6.3 Word2Vec Storage Memory Requirements

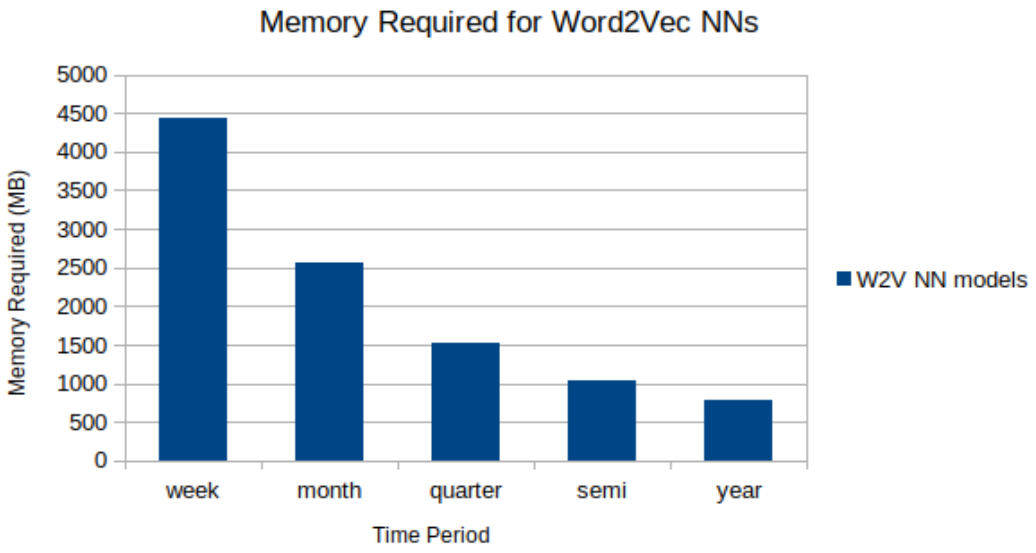

Figure A-6-11: Memory Required for the Word2Vec Neural Network Models
The above graph is a plot of the data in Table 5.3, i.e. the memory required to store the trained NN models in memory.

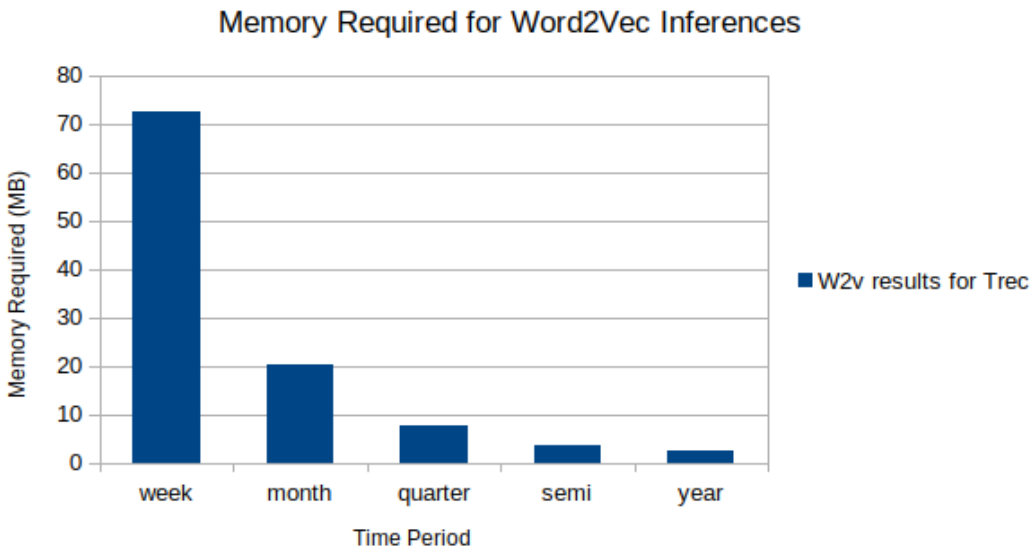

Figure A-6-12: Memory Required for the Word2Vec User Inferences
The above graph is a plot of the data in Table 5.3, i.e. the memory required to store the Word2Vec relevance judgements in memory.

A.7 Results from TREC

The below two subsections list the results obtained from Trec Eval for both Word2Vec and TRI.[105]

A.7.1 Temporal Random Indexing Results from TREC

	Month	**Quarter**	**Semi Year**	**Year**
num_q	37608	13005	6903	3485
num_ret	376072	130050	65392	.4842
num_rel	319040	113701	61031	31055
num_rel_ret	26254	2044	1333	160
map	0.0359	0.0077	0.0097	0.0026
gm_map	0.0004	0.0000	0.0000	0.0000
Rprec	0.0741	0.0169	0.0209	0.0049
bpref	0.0854	0.0186	0.0230	0.0053
recip_rank	0.1743	0.0507	0.0590	0.0198
iprec_at_recall_0.00	0.1836	0.0516	0.0607	0.0198
iprec_at_recall_0.10	0.1836	0.0516	0.0607	0.0198
iprec_at_recall_0.20	0.0746	0.0106	0.0148	0.0026
iprec_at_recall_0.30	0.0329	0.0032	0.0053	0.0001
iprec_at_recall_0.40	0.0156	0.0012	0.0020	0.0000
iprec_at_recall_0.50	0.0105	0.0009	0.0011	0.0000
iprec_at_recall_0.60	0.0031	0.0003	0.0006	0.0000
iprec_at_recall_0.70	0.0011	0.0000	0.0004	0.0000
iprec_at_recall_0.80	0.0001	0.0000	0.0000	0.0000
iprec_at_recall_0.90	0.0000	0.0000	0.0000	0.0000
iprec_at_recall_1.00	0.0000	0.0000	0.0000	0.0000

[105] It should be noted here that the metrics for the *week* time period could not be generated from Trec Eval due to an error. The error could not be resolved since there was insufficient time to do so. (TREC ERROR: *trec_eval: No queries with both results and relevance info*)

P_5	0.0794	0.0196	0.0243	0.0064
P_10	0.0698	0.0157	0.0193	0.0046
P_15	0.0465	0.0105	0.0129	0.0031
P_20	0.0349	0.0079	0.0097	0.0023
P_30	0.0233	0.0052	0.0064	0.0015
P_100	0.0070	0.0016	0.0019	0.0005
P_200	0.0035	0.0008	0.0010	0.0002
P_500	0.0014	0.0003	0.0004	0.0001
P_1000	0.0007	0.0002	0.0002	0.0000

A.7.2 Word2Vec Results from TREC

	Week	**Month**	**Quarter**	**Semi Year**	**Year**
num_q	134988	37605	13364	6906	3468
num_ret	1349880	346050	133640	69060	34680
num_rel	1082528	319009	116648	61050	30913
num_rel_ret	23167	2497	470	48	13
map	0.0218	0.0082	0.0040	0.0006	0.0002
gm_map	0.0000	0.0000	0.0000	0.0000	0.0000
Rprec	0.0227	0.0086	0.0043	0.0008	0.0004
bpref	0.0233	0.0088	0.0045	0.0010	0.0004
recip_rank	0.1581	0.0607	0.0301	0.0042	0.0017
iprec_at_recall_0.00	0.01581	0.0607	0.0301	0.0042	0.0017
iprec_at_recall_0.10	0.1581	0.0607	0.0010	0.0004	0.0017
iprec_at_recall_0.20	0.0068	0.0025	0.0001	0.0000	0.0000
iprec_at_recall_0.30	0.0012	0.0003	0.0001	0.0000	0.0000
iprec_at_recall_0.40	0.0005	0.0001	0.0000	0.0000	0.0000
iprec_at_recall_0.50	0.0002	0.0000	0.0000	0.0000	0.0000
iprec_at_recall_0.60	0.0001	0.0000	0.0000	0.0000	0.0000

iprec_at_recall_0.70	0.0000	0.0000	0.0000	0.0000	0.0000
iprec_at_recall_0.80	0.0000	0.0000	0.0000	0.0000	0.0000
iprec_at_recall_0.90	0.0000	0.0000	0.0000	0.0000	0.0000
iprec_at_recall_1.00	0.0000	0.0000	0.0000	0.0000	0.0000
P_5	0.0328	0.0126	0.0063	0.0011	0.0006
P_10	0.0172	0.0066	0.0035	0.0007	0.0004
P_15	0.0114	0.0044	0.0023	0.0005	0.0002
P_20	0.0086	0.0033	0.0018	0.0003	0.0002
P_30	0.0057	0.0022	0.0012	0.0002	0.0001
P_100	0.0017	0.0007	0.0004	0.0001	0.0000
P_200	0.0009	0.0003	0.0002	0.0000	0.0000
P_500	0.0003	0.0001	0.0001	0.0000	0.0000
P_1000	0.0002	0.0001	0.0000	0.0000	0.0000

Bibliography

[1] Oxford Dictionaries, “user profile | Definition of user profile in English by Oxford Dictionaries,” Oxford Dictionaries, 2018. [Online]. Available: https://en.oxforddictionaries.com/definition/user_profile. [Accessed 15 March 2019].

[2] S. Kanoje, S. Girase and D. Mukhopadhyay, “User Profiling Trends, Techniques and Applications,” *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, vol. 1, no. 1, 2014.

[3] A. Bonneville-Roussy, P. Rentfrow, K. Xu and J. Potter, “Music Through the Ages: Trends in Musical Engagement and Preferences From Adolescence Through Middle Adulthood,” *Journal of personality and social psychology*, vol. 105, no. 4, pp. 703-717, 2013.

[4] R. Krikorian, “How Many are There? | Slideshare,” 11 September 2010. [Online]. Available: https://www.slideshare.net/raffikrikorian/twitter-by-the-numbers/4-How_many_are_there. [Accessed 15 March 2019].

[5] Miniwatts Marketing Group, “World Internet Users Statistics and 2019 World Population Stats,” Miniwatts Marketing Group, [Online]. Available: https://www.internetworldstats.com/stats.htm. [Accessed 15 March 2019].

[6] D. J. Power, C. Heavin, J. McDermott and M. Daly, “Defining business analytics: an empirical approach,” *Journal of Business Analytics*, vol. 1, no. 1, pp. 40-53, 2018.

[7] User Modeling Inc., “User Modeling,” User Modeling Inc., 2019. [Online]. Available: https://www.um.org/. [Accessed 15 March 2019].

[8] D. Quercia, M. Kosinski, D. Stillwell and J. Crowcroft, “Our Twitter Profiles, Our Selves: Predicting Personality with Twitter,” in *Third International Conference on Privacy, Security, Risk and Trust*, 2011.

[9] R. Wald, T. Khoshgoftaar and C. Sumner, “Machine prediction of personality from Facebook profiles,” in *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, Las Vegas, NV, USA, 2012.

[10] S. C. Matz, G. M. Kosinski and D. J. S. Nave, “Psychological targeting as an effective approach to digital mass persuasion,” PNAS, 2017.

[11] T. K. Landauer, P. W. Foltz and D. Laham, “Introduction to Latent Semantic Analysis.,” Discourse Processes, 1998.

[12] Z. Xu, L. Ru, L. Xiang and Q. Yang, “Discovering User Interest on Twitter with a Modified Author-Topic Model,” in *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011.

[13] C. Wagner, V. Liao, P. Pirolli, L. Nelson and M. Strohmaier, “It's Not in Their Tweets: Modeling Topical Expertise of Twitter Users,” in *International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, 2012.

[14] Y. Xu, *User Expertise Modelling Using Social Network Data*, Dublin, Ireland: Trinity College Dublin, 2019.

[15] R. Jiamthapthaksin and T. H. Aung, “User preferences profiling based on user behaviors on Facebook page categories,” in *2017 9th International Conference on Knowledge and Smart Technology (KST)*, 2017.

[16] D. Tchuente, M. Canut, N. Baptiste-Jessel, A. Péninou and F. Sedes, “A Community Based Algorithm for Deriving Users' Profiles from Egocentrics Networks,” in *2012*

IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012.

[17] M. Kosinski, *Statement on Cambridge Analytica*, Stanford University, 2018.

[18] EUGDPR, "EUGDPR - Information Portal," EUGDPR, August 2018. [Online]. Available: https://eugdpr.org/the-regulation/. [Accessed 15 March 2019].

[19] Official Journal of the European Union, "EUR-Lex - 32016R0679 - EN - EUR-Lex," 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A32016R0679. [Accessed 15 March 2019].

[20] A. Abdel-Hafez and Y. Xu, "A Survey of User Modelling in Social Media Websites," *Computer and Information Science,* vol. 6, no. 2, pp. 59-71, 2013.

[21] A. Kobsa, "Generic User Modeling Systems," *User Modeling and User-Adaptive Systems,* vol. 11, no. 1-2, pp. 49-63, 2001.

[22] S. Schiffino and A. Amandi, "Intelligent User Profiling," in *Artificial Intelligence 5640:193-216*, 2009.

[23] European Data Protection Supervisor, "Data Protection | European Data Protection Supervisor," European Data Protection Supervisor, [Online]. Available: https://edps.europa.eu/data-protection/data-protection_en. [Accessed 18 March 2019].

[24] F. Erlandsson, M. Boldt and H. Johnson, "Privacy Threats Related to User Profiling in Online Social Networks," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, Amsterdam, Netherlands, 2012.

[25] B. Krishnamurthy and C. E. Wills, “Characterizing Privacy in Online Social Networks,” in *Proceedings of the first workshop on Online social networks (WOSN '08)*, New York, NY, USA, 2008.

[26] A. Narayanan and V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, Oakland, CA, USA, 2008.

[27] I. Dickson and D. Reynolds, “User Profiling with Privacy: A Framework for Adaptive Information Agents,” in *Intelligent Information Agents*, Springer-Verlag, Berlin, Heidelberg, 2003.

[28] F. Holz and S. Teresniak, “Towards Automatic Detection and Tracking of Topic Change,” *Computational Linguistics and Intelligent Text Processing*, pp. 327-339, 2010.

[29] C. Zhang, F. Masseglia and X. Zhang, “Modeling and Clustering Users with Evolving Profiles in Usage Streams,” in *2012 19th International Symposium on Temporal Representation and Reasoning*, Leicester, UK, 2012.

[30] S. Liang, Z. Ren, Y. Zhao, J. Ma, E. Yilmaz and M. De Rijke, “Inferring Dynamic User Interests in Streams of Short Texts for User Clustering,” *ACM Transactions on Information Systems* , vol. 36, no. 1, 2017.

[31] S. Liang, X. Zhang, Z. Ren and E. Kanoulas, “Dynamic Embeddings for User Profiling in Twitter,” KDD2018, 2018.

[32] L. Wendlandt, J. K. Kummerfeld and R. Mihalcea, “Factors Influencing the Surprising Instability of Word Embeddings,” *CoRR*, vol. 1804.09692, 2019.

[33] S. Deerwester, S. T. Dumais, G. W. Furnas, L. T. K. and R. Harshman, “Indexing by Latent Semantic Analysis,” *Journal of the American Society for Information Science*, pp. 391-407, 1990.

[34] J. R. Bellegarda, “Latent Semantic Mapping: Principles and Applications,” Morgan & Claypool, 2007.

[35] M. Sahlgren, “An introduction to random indexing,” in *Methods and Applications of Semantic Indexing Workshop*, 2004.

[36] A. Mirzal, “The limitation of the SVD for latent semantic indexing.,” in *IEEE International Conference on Control Systems, Computing and Engineering*, Mindeb, 2013.

[37] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Cornell University, 2013.

[38] P. Kanerva, “Analogy as a basis of computation,” in *Computing with Large Random Patterns*, Foundations of Real-World Intelligence, 2001, pp. 254-272.

[39] P. Kanerva, “Foundations of Real-World Intelligence,” in *Foundations of Real-World Intelligence*, CSLI Publications, 2001, pp. 1-30.

[40] A. Fonseca Bruzón, A. López-López and J. Medina Pagola, “Exploring Random Indexing for Profile Learning,” in *International Workshop on Future and Emerging Trends in Language Technology*, Seville, Spain, 2016.

[41] J. Pennington, R. Socher and C. D. Manning, “Glove: Global Vectors for Word Representation,” *EMNLP*, vol. 14, pp. 1532-1543, 2014.

[42] N. Aida, S. C. Meylan and T. L. Griffiths, “Evaluating Vector-Space Models of Word Representation, or, The Unreasonable Effectiveness of Counting Words Near Other Words,” in *CogSci*, 2017.

[43] keitakurita, “ML Explained - Machine Learning for practitioners,” mlexplained.com, 12 March 2018. [Online]. Available: https://mlexplained.com/author/admin/. [Accessed 19 March 2019].

[44] D. Selivanov, “Data Science Notes,” 12 January 2015. [Online]. Available: http://dsnotes.com/post/glove-enwiki/. [Accessed 19 March 2019].

[45] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, “Bag of Tricks for Efficient Text Classification,” Facebook AI Research, 2016.

[46] V. Zolotov and D. Kung, “Analysis and Optimization of FastText Linear Text Classifier,” IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, 2017.

[47] X. Zhang and Y. LeCun, “Text understanding from scratch,” *CoRR*, 2016.

[48] Y. Xiao and K. Cho, “Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers,” *CoRR*, 2016.

[49] A. Conneau, H. Schwenk, L. Barrault and Y. LeCun, “Very deep convolutional networks for natural language processing,” *CoRR*, 2016.

[50] Z. Yao, Y. Sun, W. Ding, N. Rao and H. Xiong, “Dynamic Word Embeddings for Evolving Semantic Discovery,” in *WSDM*, Marina Del Rey, CA, USA, 2018.

[51] D. Jurgens and K. Stevens, “Event Detections in Blogs using Temporal Random Indexing,” 2009.

[52] P. Basile, A. Caputo and G. Semeraro, “Temporal Random Indexing: a Tool for Analysing Word Meaning Variations in News,” in *Proceedings of the NewsIR'16 Workshop at ECIR*, Padua, Italy, 2016.

[53] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” NIPS, 2013.

[54] Y. Kim, T. Chiu, K. Hanaki, D. Hegde and S. Petrov, “Temporal Analysis of Language through Neural Language Models,” 2014.

[55] Y. Fang and A. Godavarthy, “Modelling the Dynamics of Personal Expertise,” *SIGIR Conference on Research & Development in Information Retrieval*, vol. 37, pp. 1107-1110, 2014.

[56] N. Blagus and S. Žitnik, “Social media comparison and analysis: The best data source for research?,” in *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, Nantes, France, 2018.

[57] Y. Wang, J. Callan and B. Zheng, “Should We Use the Sample? Analyzing Datasets Sampled from Twitter’s Stream API,” *ACM Transactions on the Web*, vol. 9, pp. 1-23, 2015.

[58] R. Abbasi, G. Rehman, J. Lee, F. M. Riaz and B. Lu, “Discovering Temporal User Interest on Twitter by Using Semantic Based Dynamic Interest Finding Model (TUT),” in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2017.

[59] H. Chen, B. Martin, C. M. Daimon and S. Maudsley, “Effective use of latent semantic indexing and computational linguistics in biological and biomedical applications,” Frontiers in Physiology, 2013.

[60] S. Liang, E. Yilmaz and E. Kanoulas, “Collaboratively Tracking Interests for User Clustering in Streams of Short Texts,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 31, no. 2, pp. 257-272, 2019.

[61] Twitter Inc., “More on restricted use cases — Twitter,” Twitter.com, [Online]. Available: https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases. [Accessed 4 March 2019].

[62] R. Hurek, “gensim: topic modelling for humans,” [Online]. Available: https://radimrehurek.com/gensim/. [Accessed 12 April 2019].

[63] A. J. Soto, C. Ryan, F. P. Silva, T. Das, J. Wolkowicz, E. E. Milios and S. Brooks, “Data Quality Challenges in Twitter Content Analysis for Informing Policy Making in Health Care,” in *Proceedings of the 51st Hawaii International Conference on System Sciences* , Waikoloa Village, Hawaii, 2018.

[64] J. Firth, “A synopsis of linguistic theory 1930-1955,” *Studies in Linguistic Analysis,* pp. 1-32, 1957.

[65] pippokill, “pippokill/tri: Temporal Random Indexing,” [Online]. Available: https://github.com/pippokill/tri. [Accessed 8 April 2019].

[66] xpo6, “List of English Stop Words - XPO6,” Arrat, 09 November 2017. [Online]. Available: http://xpo6.com/list-of-english-stop-words/. [Accessed 12 April 2019].

[67] M. Röder, A. Both and A. Hinneburg, “Exploring the Space of Topic Coherence Measures,” in *WSDM '15 Proceedings of the Eighth ACM International conference on Web Search and Data Mining,* Shanghai, China, 2015.

[68] S. Liang, “sliang1/UCT-Dataset-BitBucket,” [Online]. Available: https://bitbucket.org/sliang1/uct-dataset/get/UCT-Dataset.zip. [Accessed 12 April 2019].

[69] H. Rubenstein and J. Goodenough, “Contextual correlates of synonymy,” vol. 8, no. 10, pp. 627-633, 1965.

[70] Z. S. Harris, “Distributional Structure,” 1954.

[71] Merriam Webster, “How many words are there in English? | Merriam-Webster,” Merriam Webster, 2018. [Online]. Available: https://www.merriam-webster.com/help/faq-how-many-english-words. [Accessed 15 March 2019].

[72] Statista.com, “Facebook Users Worldwide 2018 | Statista,” Statista.com, January 2019. [Online]. Available: https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. [Accessed 16 March 2019].

[73] T. Schnabel, I. Labutov, D. M. Mimno and T. Joachims, “Evaluation Methods for Unsupervised Word Embeddings,” *EMNLP*, pp. 298-307, 2015.

[74] L. R. Goldberg, “Language and Individual Differences: The Search for Universals in Personality Lexicons,” in *Review of Personality and Social Psychology*, Beverly Hills, CA, 1981.

[75] F. Galton, “Measurement of Character,” *Fortnightly Review*, vol. 36, pp. 179-185, 1884.

[76] F. Cai, S. Liang and M. de Rijke, “Prefix-Adaptive and Time-Sensitive Personalized Query Auto Completion,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, 2015.

[77] The Apache Software Foundation, “Apache Tika - Apache Tika,” [Online]. Available: https://tika.apache.org/. [Accessed 7 April 2019].

[78] The Apache Software Foundation, “OptimaizeLangDetector (Apache Tika 1.16 API),” [Online]. Available: https://tika.apache.org/1.16/api/org/apache/tika/langdetect/OptimaizeLangDetector.html. [Accessed 7 April 2019].

[79] S. Tihon, “Stanford.NLP.CoreNLP - Stanford.NLP.NET,” [Online]. Available: http://sergey-tihon.github.io/Stanford.NLP.NET/samples/CoreNLP.html. [Accessed 8 April 2019].

[80] University of Pittsburgh Office of Technology Management, “semanticvectors/semanticvectors: SemanticVectors creates semantic WordSpace models from free natural language text.,” [Online]. Available: https://github.com/semanticvectors/semanticvectors. [Accessed 10 April 2019].

[81] J. R. Firth, “The Technique of Semantics,” London: Oxford University Press, London, UK, 1935.

[82] Facebook Inc., “Facebook Pages for marketing your business | Facebook Business,” Facebook.com, 2019. [Online]. Available: https://www.facebook.com/business/pages. [Accessed 19 March 2019].

[83] InternetLiveStats.com, “Number of Internet Users (2016) - Internet Live Stats,” InternetLiveStats.com, [Online]. Available: http://www.internetlivestats.com/internet-users/. [Accessed 12 April 2019].

[84] internetlivestats.com, “Total number of Websites - Internet Live Stats,” internetlivestats.com, [Online]. Available: http://www.internetlivestats.com/total-number-of-websites/. [Accessed 12 April 2019].

[85] statista.com, “Global Big Data market size 2011-2027 | Statista,” statista.com, [Online]. Available: https://www.statista.com/statistics/254266/global-big-data-market-forecast/. [Accessed 12 April 2019].

[86] Guardian News & Media Limited, “Facebook fined for data breaches in Cambridge Analytica scandal | Technology | The Guardian,” theguardian.com, [Online]. Available: https://www.theguardian.com/technology/2018/jul/11/facebook-fined-for-data-breaches-in-cambridge-analytica-scandal. [Accessed 12 April 2019].

[87] The New York Times Company, “Facebook and Cambridge Analytica: What You Need to Know as Fallout Widens - The New York Times,” The New York Times Company, [Online]. Available: https://www.nytimes.com/2018/03/19/technology/facebook-cambridge-analytica-explained.html. [Accessed 12 April 2019].

[88] The New York Times Company, “Google Is Fined $57 Million Under Europe’s Data Privacy Law - The New York Times,” The New York Times Company, [Online]. Available: https://www.nytimes.com/2019/01/21/technology/google-europe-gdpr-fine.html. [Accessed 12 April 2019].

[89] F. Sandin, B. Emrulo and M. Sahlgren, “Random Indexing of Multidimensional Data,” *Knowledge and Information Systems,* vol. 52, no. 1, pp. 267-290, 2017.

[90] M. Tear, M. Thompson and J. Tangen, “The Importance of Ground Truth: An Open-Source Biometric Repository,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting.,* vol. 54, 2010.

[91] K. Ganesan, “Gensim Word2Vec Tutorial - Full Working Example | Kavita Ganesan,” 2018 January 30. [Online]. Available: http://kavita-ganesan.com/gensim-word2vec-tutorial-starter-code/#.XKN7gphKhPY. [Accessed 12 April 2019].

[92] RapidAPI, " Wayback Machine API Documentation (community) | RapidAPI," [Online]. Available: https://rapidapi.com/community/api/wayback-machine. [Accessed 11 April 2019].

[93] D. Zhu and H. Dreher, "Personalized information retrieval in digital ecosystems," in *2nd IEEE International Conference on Digital Ecosystems and Technologies*, Phitsanulok, 2008.

[94] The New York Times, "Mark Zuckerberg Testimony: Senators Question Facebook's Commitment to Privacy - The New York Times," 10 April 2018. [Online]. Available: https://www.nytimes.com/2018/04/10/us/politics/mark-zuckerberg-testimony.html. [Accessed 4 March 2019].

[95] MongoDB Inc., "Advantages Of NoSQL | MongoDB," 2019. [Online]. Available: https://www.mongodb.com/scale/advantages-of-nosql. [Accessed 6 April 2019].

[96] National Institute of Standards and Technology, "Text REtrieval Conference (TREC) trec_eval," National Institute of Standards and Technology, [Online]. Available: https://trec.nist.gov/trec_eval/. [Accessed 6 April 2019].