BIRMINGHAM CITY UNIVERSITY

DOCTORAL THESIS

Leveraging Temporal Word Embeddings for the Detection of Scientific Trends

Author:
Amna DRIDI

Supervisors:
Pr. Mohamed Medhat GABER
Dr. Atif AZAD
Dr. Jagdev BHOGAL

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Data Analytics and Artificial Intelligence
School of Computing and Digital Technology

July 7, 2021

Declaration of Authorship

I, Amna DRIDI, declare that this thesis titled, "Leveraging Temporal Word Embeddings for the Detection of Scientific Trends" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Awareness by itself is not enough: it must be joined by mastery. We need gradually to develop a steering ability to keep ourselves from slipping mechanically into this or that sub-personality. Thus we become able to identify with each part of our being as we wish. We can have more choice. It is the difference between being impotently transported by a roller coaster and, instead, driving a car and being able to choose which way to go and for what purpose to make the journey."

Piero Ferrucci — *What We May Be (1982)*

BIRMINGHAM CITY UNIVERSITY

Abstract

Faculty of Computing, Engineering and the Built Environment
School of Computing and Digital Technology

Doctor of Philosophy

Leveraging Temporal Word Embeddings for the Detection of Scientific Trends

by Amna DRIDI

Tracking the dynamics of science and early detection of the emerging research trends could potentially revolutionise the way research is done. For this reason, computational history of science and trend analysis have become an important area in academia and industry. This is due to the significant implications for research funding and public policy. The literature presents several emerging approaches to detecting new research trends. Most of these approaches rely mainly on citation counting. While citations have been widely used as indicators of emerging research topics, they pose several limitations. Most importantly, citations can take months or even years to progress and then to reveal trends. Furthermore, they fail to dig into the paper content.

To overcome this problem, this thesis leverages a natural language processing method – namely *temporal word embeddings* – that learns semantic and syntactic relations among words over time. The principle objective of this method is to study the change in pairwise similarities between pairs of scientific keywords over time, which helps to track the dynamics of science and detect the emerging scientific trends. To this end, this thesis proposes a methodological approach to tune the hyper-parameters of *word2vec* – the word embedding technique used in this thesis – within the scientific text. Then, it provides a suite of novel approaches that aim to perform the computational history of science by detecting the emerging scientific trends and tracking the dynamics of science. The detection of the emerging scientific trends is performed through the two approaches *Hist2vec* and *Leap2Trend*. These two approaches are, respectively, devoted to the detection of *converging keywords* and *contextualising keywords*. On the other hand, the dynamics of science is performed by *Vec2Dynamics* that tracks the evolvement of semantic neighborhood of keywords over time.

All of the proposed approaches have been applied to the area of *machine learning* and validated against different gold standards. The obtained results reveal the effectiveness of the proposed approaches to detect trends and track the dynamics of science. More specifically, *Hist2vec* strongly correlates with citation counts with 100% Spearman's positive correlation. Additionally, *Leap2Trend* performs with more than 80% accuracy and 90% precision in detecting emerging trends. Also, *Vec2Dynamics* shows great potential to trace the history of machine learning literature exactly as the machine learning timeline does. Such significant findings evidence the utility of the proposed approaches for performing the computational history of science.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my director of studies Professor Mohamed Gaber, who has the substance of a genius: he expertly guided me through my doctoral studies. His support and encouragement were unparalleled since the very first steps of this PhD journey, and over the course of the past three and half years. His unwavering enthusiasm for the research topic kept me constantly engaged with my research, and his personal generosity helped me make my PhD time enjoyable. Without his persistent help and constant feedback, this PhD would not have been achievable. I am incredibly grateful for his willingness and generosity to share his time and remarkable wealth of knowledge as well as for his punctual and extensive comments on my work. He has also provided me with invaluable teaching and reviewing opportunities. I also remain indebted for his understanding and support during the times when I was really down and depressed due to various personal challenges. I am convinced that there exists no better supervisor, and it has truly been an honor and a pleasure working with him.

My gratitude is also extended to my second supervisor Dr. Atif Azad for his academic supervision and personal support. He has been an unfailing source of encouragement, advice and reassurance. He has mainly provided valuable assistance with scientific writing. Dr. Atif's enthusiasm for my topic, was essential in helping enriching our meetings that never feel like meetings. Our meetings feel like friends sharing things such as advice and stories. My gratitude would not be complete without thanking Dr. Jagdev Bhogal; my third supervisor, who has been a continuing source of encouragement and optimism throughout. Every time we talk, she never misses giving me compliments and making me feel she enjoys advising me, this means a lot for me.

My acknowledgements also go to Birmingham City University for the award of the PhD scholarship, which enabled me to undertake my doctoral studies. Without this financial support, the completion of this degree would not have been possible. A very special thank you goes out to the Executive Dean Prof Hanifa Shah for her tolerance and encouraging words every time she sees me. I gratefully acknowledge her relevant support during my sickness at the very beginning of my PhD. I also welcome this opportunity to thank the Doctoral Research College officers, mainly Andrea Mondokova for her availability and assistance.

My sincere appreciation goes out to Data Analytics & Artificial Intelligence (DAAI) team members for a cherished time spent together in the office and during DAAI seminars. I am especially grateful to Dr. Shadi Basurra, Dr. Zahraa Abdallah and Dr. Mariam Adedoyin-Olowe; the members of my PhD progression panel for two successive years, for their constructive feedback and the interest they have shown to my research topic.

I must also thank Dr. Imed Romdhani – the associate professor at Edinburgh Napier University – for his continued encouragement and support throughout my doctoral journey. His calls and valuable advice were a continuing source of optimism.

My special thanks are due to my colleagues, lab-mates and friends: Besher for all his personal generosity and willingness to help in whatever way, for all his attention and advice mainly during my pregnancy; Aliyu for being there to listen when I needed an ear and for boosting my confidence, that is something I really needed and appreciated; Hussein for his nice attitude and the invaluable time sharing the teaching experience challenges and laughter; Khadija for all her prayers and calls –

despite my own limited devotion to calls; Julian, Zakaria, Hansi and Lorraine for their friendship and the warmth they extended to me during my PhD that have made my time at BCU a wonderful time. A very special thank you to Salamah – friend, 'brother', colleague and lecturer – for always being so supportive and helpful in numerous ways since my first day at BCU, with whom I have shared laughter, frustration and companionship, and with whom I have also shared the office, the home-made foods and all canteen times. All the ways he went twice – just because I was walking in slow motion with my big belly – are unforgettable. I owe a great deal to his time and unconditional help, especially during the lockdown in whatever way he could during this challenging period.

Heartfelt thanks go to Nabila and Vanessa – housemates and friends – who have been an unstinting source of support throughout my stay in the UK, especially during the challenging time of my sickness and pregnancy. A special thank you goes also to Asma for her time and support during my maternity time, especially during the lockdown; what she has done means a lot for me. I think it is also essential that I thank my long term friends Farida, Elhem, Neila, Boudour, Maram and Asma for their continuous calls and texts that I usually miss, and for being kind and supportive to me over the last several years.

None of this could have happened without my family, whose great love and continuous prayers were essential in my determination to find and realise my potential. To my father Lakhdhar and my mother Shadia and my brothers: Mohamed; Ahmed and Hazem, I am forever grateful for the unceasing support and attention. Without their tremendous understanding and encouragement, I would not have had the courage to embark on this journey in the first place. Most importantly, I am indebted to my mum Shadia for always believing in me and encouraging me to follow my dreams. I am especially grateful for helping me with my baby through the months of writing and correction, for her continuous encouragement and the energy she is offering everytime I feel down. It is also dedicated to my grandmother, my uncle Mohamed Bouzaiene, aunties, cousins Wassim, Ameni and Khaled and sister-in-low Saida for supporting me spiritually throughout my life. It is amazing to have family close by so far away from home.

To my husband Mustafa, who – despite all the challenging time – led me to an understanding of some of the more subtle challenges to my ability to thrive.

And finally, to little Mahdi who has given my life new meaning and also challenged me like never before. With nine months in my belly and ten months in my arms, sharing accordingly the half journey of my PhD, thank you for being such a good little boy, and making it possible for me to complete what I started.

Contents

List of Figures

List of Tables

List of Abbreviations

ACPM	**A**dvanced **C**lique **P**ercolation **M**ethod
ANN	**A**rtificial **N**eural **N**etwork
ASE	**A**ction **S**cience **E**xplorer
CBOW	**C**ontinuous **B**ag of **W**ords
CORE	**A**ustralian **C**omputer **R**esearch and **E**ducation conference ranking
CS	**C**omputer **S**cience
DCA	**D**ocument **C**itation **A**nalysis
DM	**D**istributional **M**emory
DNNs	**D**eep **N**eural **N**etworks
DS	**D**istributional **S**emantics
DSM	**D**istributional **S**emantic **M**odel
DV	**D**ependency **V**ectors
FoS	**F**ield **o**f **S**tudy
GloVe	**G**lobal **V**ectors
HAL	**H**yperspace **A**nalogue of **L**anguage
HiDEx	**H**igh **D**imensional **E**xplorer
HMM	**H**idden **M**arkov **M**odel
IS	**I**nformation **s**cience
***k*-NN**	**k** **N**earest **N**eighbors
LDA	**L**atent **D**irichlet **A**llocation
LRA	**L**atent **R**elational **A**nalysis
LREC	**L**anguage **R**esources **E**valuation **C**onference
LSTM	**L**ong **S**hort **T**erm **M**emory
LREC	**L**anguage **R**esources **E**valuation **C**onference
MAG	**M**icrosoft **A**cademic **G**raph
MICCAI	**M**edical **I**mage and **C**omputer **A**ssisted **I**ntervention
ML	**M**achine **L**earning
NIPS	**N**eural **I**nformation **P**rocessing **S**ystems
NLP	**N**atural **L**anguage **P**rpcessing
R & D	**R**esearch and **D**evelopment
RIDP	**R**eference **I**njection based **D**ouble **D**amping **P**age
SG	**S**kip **G**ram model
SNA	**So**ocial **N**etwork **A**nalysis
t-SNE	**t**- distributed **S**toachstic **N**eighbor **E**mbedding
TWE	**T**emporal **W**ord **E**mbedding
VSM	**V**ector **S**pace **M**odel
WE	**W**ord **E**mbedding
WMD-CCA	**W**ord **M**over's **D**istance **C**onstructive **C**overing **A**lgorithm
WSD	**W**ord **S**ense **D**isambiguation
5**V**	**V**olume **V**ariety **V**elocity **V**alue and **V**eracity

To my little son Mahdi,
You have made me stronger, better and more blissed out than I could have ever imagined – I cannot offer a better gift in your first birthday. I love you to the moon and back.

– Amna

Chapter 1

Introduction

> "Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."
>
> — Lewis Carroll. (1832-1898), *Alice in Wonderland*

1.1 Preamble

Due to the easily accessible reservoir of data, researchers and scholars have an increased need for a deeper understanding of the structure and dynamics of science. In other words, sophisticated techniques and tools are highly solicited to help researchers to learn better about knowledge production processes, curate insights from scholarly data and speculate upcoming research topics. This PhD thesis proposes novel approaches towards a fine-grained computational history of science. These approaches follow different methodologies – all of them aiming to track the dynamics of science and detect the emerging scientific trends.

This chapter introduces the motivation behind the research problem addressed in this thesis related to scholarly data mining – the field that studies scholarly data, detailed in Chapter 2 – and trend analysis – the related sub-field that interests the study of the upcoming topics. Furthermore, it details the problem statement and along with it raises the corresponding research questions. Afterward, it highlights the main contributions, and it concludes by setting the outline of this thesis.

1.2 Motivation

With the vast increase of research works undertaken in academia and industry and the widespread use of scholarly networks and digital libraries, we now have access to an abundant academic resources such as books, peer-reviewed articles, conference papers and websites from authoritative organisations and institutions. These resources are ranging from more than 300k to almost 400 million offered by 28 search systems in 2020 (Gusenbauer and Haddaway, 2020). In a further study performed by (Bornmann and Mutz, 2014) on the growth of science, the two bibliometric analysts have found the real rate of the global scientific output closer to 8-9% in 2012 each year, which is equivalent to a doubling of global scientific output roughly every nine years.

As a consequence of this increasing volume of scholarly data and the growth rate of scientific output, the extraction of useful knowledge and the understanding of the

structure and dynamics of science are hampered (Gaber, 2010). This has recently led to the emergence of *scholarly data mining* as an important research field, facing new challenges due to the typical nature of science, considering the complexity of the academic landscape and the 5V feature (Volume, Variety, Velocity, and Value) of scholarly data (Kaisler et al., 2013; Feng et al., 2017).

Currently, the main problem that the researchers and scholars are facing is not simply obtaining any useful information from this accessible reservoir of data. It is rather understanding the structure of the scholarly communication, and tracking the dynamics of science. This could provide better academic services for scholars and researchers. Nevertheless, this is not a trivial task since scholarly data is very different and usually includes some special features such as (i) the complexity drawn from the fact that it involves various entities (papers, authors, journals) and relationships among these entities and (ii) the veracity, which comes from author disambiguation and deduplication (Ferreira et al., 2012).

The extraction of useful knowledge from this amount of data is essential to provide support not only to scholars on their understanding of the rules and laws of science (Feng et al., 2017), but also to governments and institutions on several decision-making processes such as policy making for fund disbursement, speculating upcoming research areas, etc.

In this regard, *scholarly data mining* has become essential for several key reasons. First, it is the availability of abundant academic resources. In addition to the scholarly documents such as papers, books, reports, *etc*, multiple associated data is available today including information about authors, citations, institutions, funds and academic networks (J. Liu et al., 2018). Furthermore, there have been several initiatives by governments and organisations to digitise academic resources in order to meet the challenges of information explosion. As a matter of fact, the scholarly communication has been revolutionised drastically over the past two decades due to the unprecedented advancement in information and communication technology (IT). For instance, IT has been used to digitise the knowledge that used to be in the conventional print form. It has then brought a revolutionary form in archiving and accessing knowledge.

Second, due to this easily accessible reservoir of data, researchers and scholars are in an elevating need for a deeper understanding of the structure and dynamics of science. Consequently, developed tools are highly demanded to assist scholars in the knowledge production processes and the prediction of the upcoming research trends.

Third, the availability of this vast amount of data about scientists' collaborations, document sharing and publications enables the evaluation of scientific impact of different entities including papers, authors and journals. The measurement of this scientific impact is deemed vital for governments and businesses for decision-making processes such as funding allocation, research gap identification, university ranking determination, tenure and recruitment decisions.

Fourth, in addition to studying the advantages of the scientific impact of available resources, it is also essential to consider the negative effects of science, or what is termed *"Bad Science"* (Goldacre, 2008). In fact, the scientific misconduct is heavily present in the scientific communities and it has different forms. For example, researchers copying others, presenting false results, or distorting of the research process by fabrication of data, text, hypothesis, or methods from another researcher's manuscript. Similarly, for citations, the citing behaviour could possibly be for non-scientific reasons (Bornmann and Daniel, 2008), such as the reputation of an author

or a journal, or other bad reasons. This could possibly mislead the identification of research trends that are based on the citation counts. Consequently, developed tools are demanded to assist scholars, researchers and organisations to understand the scientific behaviour and uncover the cases of *Bad Science*.

Finally, going beyond the study of the scientific impact, scholarly data analysis also promotes the understanding of human social activities. It provides sociologists with valuable data to observe researcher interactions and community formation. It also allows countries to evaluate the impact of institutions or scientists to allocate resources. Overall, scholarly data mining contributes to the *Science of Science* (Fortunato et al., 2018; Light et al., 2014) that advances our understanding of the structure and dynamics of science.

According to the literature, there have not been enough attempts to closely study the approaches for scholarly data analysis. An effort in this direction could reveal new branches for research in this area. One of the main branches of scholarly data mining is to study how research topics evolve over time and to track emerging topics and trends. In the literature, this task is termed as *trend analysis* (An et al., 2017; Dridi et al., 2019b; Hou et al., 2018; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017).

Trend analysis has received considerable interest in the past few years, because finding a research trend is key to finding a niche in a particular field of interest, especially for those new to this field. The main goal of trend analysis is to reveal hidden trends within these vast resources, such as research trend evolution and community dynamics (Feng et al., 2017).

The emerging research addressing trend analysis can be categorised into three categories: *bibliometrics-based approaches* (An et al., 2017; Hou et al., 2018; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017) that are based on social network analysis, citation and co-citation analysis; *content-based approaches* (Dridi et al., 2019a; Dridi et al., 2019b; Weismayer and Pezenka, 2017) that treat entities – essentially keywords – reflecting the paper content (Weismayer and Pezenka, 2017) or dig deeply into the paper content and study the associations between keywords (Dridi et al., 2019a; Dridi et al., 2019b); and *hybrid approaches* (Effendy and Yap, 2017; Hoonlor et al., 2013) that combine both citation and content.

Despite the relatively interesting body of literature on trend analysis, there are still open issues to tackle. Specifically, the following issues: the limitations of citation counts and the drawbacks of the natural language processing techniques used to model the scientific content. Consequently, it becomes essential to closely study the approaches for trend analysis, explore the techniques involved in mining scholarly data, and find the most suitable techniques to improve the understanding of the structure and dynamics of science, and accordingly reveal hidden trends within the vast quantity of available resources.

1.3 Problem Statement

Over the past few years, the computational history of science – as a part of scholarly data mining/analysis (Feng et al., 2017) – has grown into a scientific research area that is increasingly being applied in different domains such as business, biomedical, and computing. The surge in interest is due to (i) the explosion of publicly available data on scholarly networks and digital libraries, and (ii) the importance of the study of scientific literature, which is continuously evolving. In fact, the recent literature

is rich in dealing with the enigmatic question of the dynamics of science (An et al., 2017; Ashton et al., 2012; Effendy and Yap, 2017; Hall et al., 2008; Hoonlor et al., 2013; Hou et al., 2018; Mortenson and Vidgen, 2016; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017). The dynamics of science does not only involve tracking topic evolvement (Zehra and Umut, 2018), but also how to predict future research trends and popular research topics, termed as *trend analysis* approaches (Feng et al., 2017).

Most of these approaches rely mainly on citation counting from papers, which have been published and consequently find clues to topic evolvement (Zehra and Umut, 2018). While citation counts are used as indicators of emerging research topics, they can take months or even years to reveal research trends. Also, they fail to dig into the paper content, which could lead to a more accurate computational history. Therefore, there is a need to shift from citation-based approaches to more fast yet accurate approaches for trend analysis that drill into the content of scholarly publications.

Despite their ability to give an overview on the appearance/disappearance and the frequencies of keywords, the statistical methods were not considered for the content analysis of scholarly publications in this thesis and in the literature in general. This is justified by the fact that the count of keywords does not reflect the semantics embedded within the scientific language. The extraction and the understanding of these semantics help to generate new knowledge. For this reason, semantic-based approaches for text mining are needed to analyse the content of scholarly publications. Following this trend, some work (Ashton et al., 2012; Bakarov et al., 2018; Li et al., 2019; Paul and Girju, 2009; Salatino et al., 2017) emerged and explored *topic models* to forecast the emergence of new research topics. While topic models intend to extract semantics by capturing document level associations among words, they fail to detect pairwise associations of keywords. This is a considerable limitation since emerging topics often start first by an increasing closeness of keywords that may lead to a merge. For instance, in the artificial intelligence field, the research topic *"deep learning"* resulted from the merge between the two keywords/topics *"machine learning"* and *"neural networks"*. For this reason, a fine-grained study of the associations between pairs of keywords is needed for an early detection of emerging trends and early uncovering of the dynamics of science. In this thesis, the *early* detection of emerging trends means detecting the trends at a very early stage; at the time of publication of new scholarly communications. In comparison with citations that need months or even years to be accumulated, the idea here is to analyse the content of available publications and early reveal the pairs of keywords that the combination is likely to result in a new trend.

The goal in this PhD project is to propose a suite of novel methods that aim to effectively and early detect the emerging scientific trends and track the dynamics of science by addressing the limitations of topic models. To do so, word embeddings (Mikolov et al., 2013c) as a natural language processing technique are leveraged in this thesis due to their abilities to detect pairwise similarities between words. Word embeddings represent words in the form of real-valued vectors such that the words that are closer in the vector space are expected to be similar in meaning. This vector representations of words can promote the automatic knowledge extraction from unstructured text. This is useful for different NLP tasks such as semantic text analogy, word-sense disambiguation, sentiment analysis, *etc.* Considering that this thesis will tackle the semantic text analogy within the scientific language, therefore any word embedding technique could be applied. In this thesis, word2vec (Mikolov

et al., 2013e) is chosen to be applied in a temporal fashion, which is termed in this thesis as *temporal word embeddings*. A detailed treatment on the choice of *word2vec* is discussed in Chapter 3, Section 3.5.4.

1.4 Research Questions

With the above-stated research problem in mind, the following research question – that represents the core research question of this thesis – is asked:

> ***How the dynamics of science can be tracked and how the emerging scientific trends can be early detected?***

Based on the core research question, three related questions can be revealed as follows:

1. ***Research Question 1: How to represent the scientific text with natural language processing techniques that help to reveal the semantics and the dynamics of scientific keywords over time?***

   To answer this question, three sub-questions have to be answered:

   - Are word embeddings – as a natural language processing technique – (namely *word2vec* (Mikolov et al., 2013e)) able to detect semantic and syntactic analogies in scientific language?
   - How the hyper-parameters of word embedding techniques can be tuned?
   - How to find/create analogy datasets as gold standard to validate the ability of word embeddings to detect semantic analogies from scientific text?

2. ***Research Question 2: How to explore the vector representation of scientific keywords in order to study their semantic shifts, and consequently perform a computational history of science?***

   To answer this question, the following sub-questions have to be answered:

   - How to detect the semantic shifts of scientific keywords over time?
   - How to perform the computational history of science?
   - How to represent the temporal dimension in an effective way to perform the computational history of science?

3. ***Research Question 3: How to evaluate the detected emerging trends and validate the obtained results on the dynamics of science?***

   To answer this question, the following five sub-questions have to be answered:

   - How to find/define gold standards related to the application areas that help to define scientific trends?
   - Which standard validation measures can be used to assess the effectiveness of the obtained results?
   - How to conduct comparative studies with existing approaches?
   - Which standards can be defined for descriptive analyses, where normative analyses are not suitable for the analysis of the computational history of science?
   - How visual analyses can be used as qualitative analyses to highlight the semantic shifts of scientific keywords over time?

1.5 Contributions

This thesis **proposes a suite of novel approaches towards a fine-grained computational history of science. These approaches follow different methodologies – all of them aiming to track the dynamics of science and detect the emerging scientific trends. To do so, a word embedding technique – namely word2vec (Mikolov et al., 2013a) – is leveraged and applied to scientific literature across time to learn the change in pairwise similarities between pairs of scientific keywords.**

To this end, three main stages are followed. **(1) The first stage is devoted to methodological studying word embeddings** – the natural language processing technique this thesis uses for the representation of scientific text – and deeply understand the embedding behavior within scientific corpora. This methodological study concerns finding the optimal hyper-parameters of word embeddings. Typically, according to the literature, most of the existing research work that used word embeddings as features computed their vector representations with a default or arbitrary choice of embedding hyper-parameters. However, these hyper-parameters are crucial to the prediction performance as they directly affect the accuracy of the generated analogies. Given that analogies can be used in hypotheses synthesis, consequently an accurate analogy will lead to a precise hypothesis. For example, *"decision tree"* is a component of *"ensemble"* and *"decision tree"* is a *"classifier"*. So, by analogy any classifier should be a component of an ensemble. While this thesis does not synthesise hypotheses, it is worth noting that having accurate analogies will regulate the accuracy of hypotheses, and the accuracy of these hypotheses will regulate the accuracy of finding trending keywords. It is now understood that the hyper-parameters of word embeddings play an important role in generating accurate analogies. For this reason, it is crucial to methodologically set them.

After testing the effectiveness of word embeddings in generating analogies in scientific text and proving their ability to accurately represent the scientific language, **(2) the second stage aims to learn word embeddings across time and use their outputs to propose different approaches towards a computational history of science.** In this thesis, the computational history of science is performed following two paths. (i) The first path is devoted to the detection of the emerging scientific trends/keywords. These trends are defined according to the flow of science as *converging keywords* or *contextualising keywords*. The **converging keywords** refer to the keywords that converge in similarity over time, which lead to the appearance of new keyword as a merge of these converging keywords. For example, *"deep learning"* is the keyword that emerged from the convergence between *"machine learning"* and *"neural networks"* at a specific point in time. Similarly, *"bioinformatics"* is the keyword that emerged from the convergence between *"biology"* and *"information engineering"*. However, **contextualising keywords** refer to the keywords that start to appear in the same context, but not necessarily merge, which lead to the emergence of a new application area or a combination of existing tools. A notable example of contextualising keywords is *"healthcare"* and *"artificial intelligence"*. This implies the application of the artificial intelligence models to the healthcare, and accordingly the emerging of this application as a scientific trend. Based on these two definitions of scientific trends (*converging keywords* and *contextualising keywords*), this thesis makes the assumption that emerging trends are defined as a pair of fast conversing keywords. It is then clear that this thesis is not looking into new terms that appear as a result other than merging existing terms. It rather focuses on the semantics behind, and provides a conceptual approach that assist with the task of early detection of

trends independently from the terminology. (ii) The second path is devoted to the study of the dynamism of keywords. This latter is performed through tracking the evolvement of the semantic neighbourhood of keywords over time, which gives a more generic view on the dynamics of science including *emerging*, *dying*, *recurrent* and *persistent* keywords.

According to the two followed paths to perform the computational history of science in this thesis, the proposed approaches are named *Hist2Vec*, *Leap2Trend* and *Vec2Dynamics*. Both *Hist2Vec* and *Leap2Trend* detect the emerging scientific trends by detecting converging and contextualising keywords, respectively. *Hist2Vec* detects the converging keywords that may result in trending keywords by computing the acceleration of similarities between keywords over successive timespans (a *timespan* is a time window that represent a fixed number of years of publications). As shown in the box (a) in Figure 1.1, the two keywords ◯ and □ are getting closer over time, until converging at a certain point in time (*timespan t*). However, going beyond the row values of similarities, *Leap2Trend* adopts the rankings of similarities and computes the ascents in ranking over different timespans to detect the trending keywords that co-occur in the same context. According to the box (b) in Figure 1.1, the pair of keywords △ and ◯ is being used in the same context in the *timespan* $(t-1)$. They are then considered contextualising keywords. While *Hist2Vec* and *Leap2Trend* tracks the trending keywords, *Vec2Dynamics* tracks the dynamics of keywords by tracking the evolvement of their semantic neighbourhood over time. The box (c) in Figure 1.1 shows an overview of the dynamics of scientific keywords. The keyword ⊖ appears only in *timespan 1* and then it totally disappears. It is considered a *dying* keyword. However, the keyword ⊗ appears in all timespans. It represents the *persistent* keywords. The keyword ⊘ appears in *timespan 1*, it disappears in next timespans, then it re-appears again in *timespan* $(t-1)$ and t. It then represents the *recurrent* keywords. Finally, the keyword ⊕ emerges in *timespan t*. It accordingly represents the *emerging* keywords.

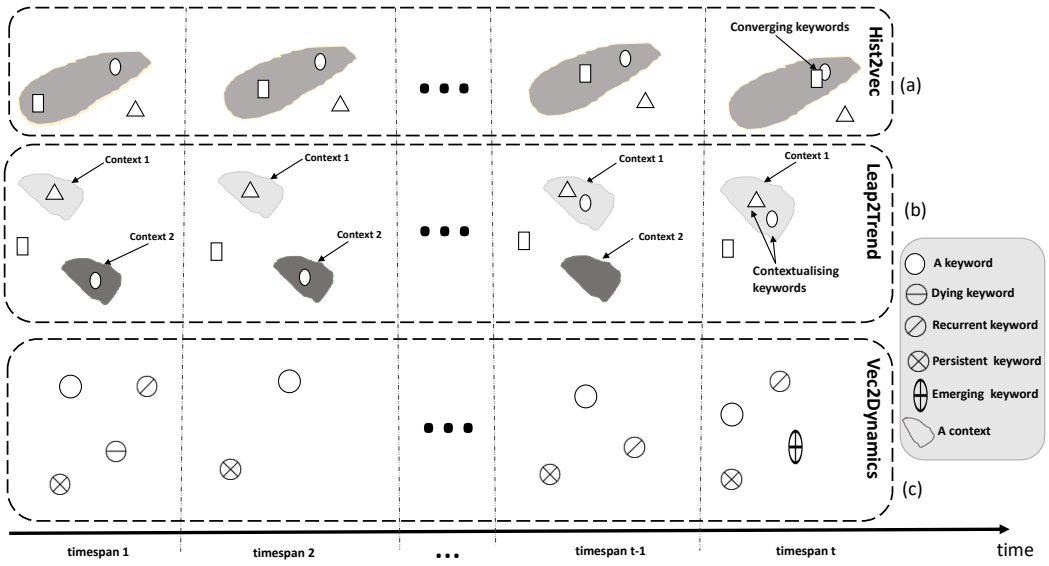


FIGURE 1.1: Overview of Hist2Vec, Leap2Trend and Vec2Dynamics

Finally, **(3) the third stage provides standards to validate the results of the proposed approaches.** Both quantitative and qualitative analyses are proposed for the

validation process.

All the proposed approaches are domain-independent and they can be applied to any scientific area. However, in this thesis, these approaches are evaluated in the area of *machine learning*. This choice is based on the authors' background, and the fact that machine learning has witnessed notable successes and growth in the recent years. Therefore, any machine learning corpora can be used to evaluate the proposed approaches. In this thesis, Neural Information Processing Systems (NIPS) – recently abbreviated as NeurIPS – corpora are used for three main reasons. First, the NIPS conference is a long standing venue in the area of machine learning with more than 30 editions. This is important for the work proposed in this thesis that requires history in publications to perform the computational history of science. Second, NIPS has a *h5-index* equal to 198 and an *impact score* equal to 33.49. These metrics show that the citation counts of the NIPS publications is weighty. This is important for this thesis because the proposed approaches will be evaluated against the citation-based approaches, where the dynamics of citation counts is essential for the comparative study. These metrics also reflect the value and the prestige of the conference. It is ranked top 2 conference in the area of machine learning and artificial intelligence, according to *Guide2Research* rankings [1]. It comes after *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, which is more specialised in computer vision applications. NIPS is then a more generic venue and covers a wide range of machine learning topics including computer vision. It can be then considered the top 1 conference in generic machine learning conferences like the *International Conference on Machine Learning (ICML)*, *International Conference on Learning Representations (ICLR)* and *European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD)* that are ranked top 6, top 17 and top 104, respectively. Finally, aligning with its reputation, the NIPS dataset is made publicly available on Kaggle[2], where the papers database is easily manageable, with the papers' features, such as the title, the abstract and the paper text, which is time-stamped. The availability of this database makes from the task of data collection an efficient task. Moreover, bioinformatics corpora are used in addition to the NIPS corpora in Chapter 6. This choice is firstly because the area of *bioinformatics* is increasingly relying on machine learning algorithms, which aligns with the application area of the thesis. Secondly, because Chapter 6 is dedicated to the detection of *contextualising keywords*, and the area of bioinformatics is one of the areas, where contextualising keywords may frequently appear. Recall that in this thesis words and keywords are used exchangeably. Words/keywords refer to either *unigrams* or *bigrams*. The choice of bigrams is justified by their frequent use in the scientific language such as *"machine learning"* or *"neural networks"*, *etc*. Recall that *trigrams* are also important and highly used in the area of machine learning such as *"artificial neural netoworks"*, *"support vector machines"*, *etc*. However, in general their abbreviations are more frequent ("ANN", "SVM"). For this reason, trigrams were not considered in this thesis. As an alternative, their abbreviations were considered as unigrams. As such, the main contributions of this thesis are fourfold:

Contribution 1: A methodological approach for tuning word embedding hyperparameters. This contribution consists of four sub-contributions:

[1] https://www.guide2research.com/topconf/machine-learning

[2] https://www.kaggle.com/benhamner/nips-papers/data

1. The use of the stability of *k-nearest neighbors (k-NN)* of word vectors as an objective to measure while learning word2vec hyper-parameters.
2. The enhancement of the standard skip-gram model (Mikolov et al., 2013c) – the word2vec architecture adopted in this thesis, which is thoroughly described in Chapter 3 – by bigrams as a method for corpus augmentation.
3. The creation of an analogy dataset for *machine learning* – the application area of this thesis – by methodologically curating Association for Computing Machinery (ACM) hierarchy and Wikipedia outline of machine learning.
4. A quantitative and qualitative evaluation of the proposed methodological approach on the NIPS corpora and the obtaining of interesting semantic relations in the area of machine learning.

The outcomes of this contribution have been published in:

"Amna Dridi and Mohamed Medhat Gaber and R. M. Atif Azad and Jagdev Bhogal (2018). **k-NN Embedding Stability for word2vec Hyper-Parametrisation in Scientific Text**. *In: Discovery Science - 21st International Conference, DS 2018, Limassol, Cyprus, October 29-31, 2018, pp. 328–343"*.

Contribution 2: Hist2Vec: A temporal word embedding approach for the detection of converging keywords. The main features of this approach are as follows:

1. A *similarity matrix* that records the similarity between pairs of frequent keywords, which are represented by their embedding vectors, over each timespan.
2. An *acceleration matrix* that computes the *acceleration* of pairs of keywords over subsequent timespans in order to detect the fast converging pairs of keywords. The acceleration represents the difference in similarities between pairs keywords over two successive timespans.
3. A qualitative and quantitative evaluation of the proposed approach with the NIPS publications between the years 1987 and 2015. The qualitative evaluation consists of *t-distributed stochastic neighbor embedding (t-SNE)* visualisations (Maaten and Hinton, 2008) that illustrate the fast acceleration between the pairs of scientific keywords. Moreover, the quantitative evaluation consists of a comparative study with citation counts approaches that returns 100% positive correlation between the citation counts and the similarities returned by the proposed approach.

The outcomes of this contribution have been published in:

"Amna Dridi and Mohamed Medhat Gaber and R. M. Atif Azad and Jagdev Bhogal (2019). **DeepHist: Towards a Deep Learning-based Computational History of Trends in the NIPS. In: International Joint Conference on Neural Networks - IJCNN 2019 Budapest, Hungary, July 14-19, 2019, pp. 1–8"**.

Contribution 3: Leap2Trend: A temporal word embedding approach for the detection of contextualising keywords. The main features of this approach are as follows:

1. Introducing a new framework for the detection of contextualising keywords, which led to the detection of new research trends at a very early stage.
2. Leveraging temporal word embedding techniques, namely word2vec, for fine-grained content analysis of scientific corpora, following two temporal paradigms: *incremental* (a sequence of time stamped corpora gradually created following a 1-year annual basis) and *sliding* (a sequence of three time stamped corpora; the corpus of the window t will contain the corpora of the timespan $(t-1, t+1)$), in order to study the impact of research history in detecting new emerging trends.
3. Applying *Leap2Trend* to real-world datasets in two research areas – machine learning and bioinformatics – which could give insights about the validity and the generalisability of the proposed approach.
4. Validating the approach using *Google Trends hits* and *Google Scholar citations* as gold standards.

The outcomes of this contribution have been published in:

"Amna Dridi and Mohamed Medhat Gaber and R. M. Atif Azad and Jagdev Bhogal (2019). **Leap2Trend: A Temporal Word Embedding Approach for Instant Detection of Emerging Scientific Trends. In: IEEE Access Journal - IEEE Volume 7, 2019, pp. 176414-176428"**.

Contribution 4: Vec2Dynamics: A temporal word embedding approach to exploring the dynamics of scientific keywords – Tracking the dynamism of scientific keywords. The main features of this approach are as follows:

1. To detect the dynamics of keywords, word vectors are learned across time. Then, based on the similarity measure between the embedding vectors of keywords, the k- nearest neighbors (k-NN) of each keyword are defined over successive timespans.
2. The change in stability of k-NN over time refers to the dynamics of keywords and accordingly the dynamics of the research area.
3. *Vec2Dynamics* is evaluated with the NIPS publications between the years 1987 and 2016.
4. Both numerical and visual methods are adopted to perform a descriptive analysis and evaluate the effectiveness of *Vec2Dynamics* in tracking the dynamics of scientific keywords.

The outcomes of this contribution have been submitted to *Springer Machine Learning journal* as:

"Amna Dridi and Mohamed Medhat Gaber and R. M. Atif Azad and Jagdev Bhogal. **Vec2Dynamics: A temporal word embedding approach to exploring the dynamics of scientific keywords"**.

1.6 Thesis Outline

The rest of this thesis is structured as follows (see Table 1.1) :

Chapter 2: Scholarly Data Mining. This chapter presents a description of the state-of-the art related to the core research area of this thesis. It introduces and discusses scholarly data mining applications following a literature-based analysis. It also reviews the methods used for scholarly data mining ranging from statistical and empirical analysis to machine learning techniques. Additionally, it highlights the areas of application of scholarly data mining. Finally, it discusses the open challenges in the area of scholarly data mining, and how this thesis contributes to address some of these challenges. The work described in this chapter has been published in:

"Amna Dridi and Mohamed Medhat Gaber and R. M. Atif Azad and Jagdev Bhogal (2020). **Scholarly data mining: A systematic review of its applications**. *In: WIREs Data Mining and Knowledge Discovery, e1395. doi.org/10.1002/widm.1395, 2021"*.

Chapter 3: Word Embedding Technique – Word2vec. This chapter presents the background chapter of the thesis. It thoroughly presents the natural language processing technique used in this thesis – *word embeddings (word2vec)*. It introduces their foundations and describes word2vec technique. Additionally, it highlights the temporal dimension given to word embeddings and summarises related work in this direction.

Chapter 4: k-NN Embedding Stability for Word2vec Hyper-parametrisation. This chapter introduces the first contribution of this thesis. It presents the methodological approach that this thesis proposed to tune word embedding hyper-parameters, which is based of the stability of k-nearest neighbors of word vectors. In addition, it details the quantitative and qualitative evaluation method that is performed to evaluate the proposed approach in the area of machine learning. This chapter, in summary, answers the first research question raised in Section 1.4, while Chapters 5, 6 and 7 provide answers to the second and third research questions raised in the same section.

Chapter 5: Hist2Vec: Detecting The Converging Keywords. This chapter presents the second contribution of this thesis. It introduces *Hist2Vec*, the temporal word embedding approach that is proposed to detect the converging scientific keywords that may lead to emerging scientific trends. This approach represents the first path that this thesis follows to perform the computational history of science by detecting the emerging scientific trends in the area of machine learning, and it is devoted to the first definition given to the emerging scientific trends – *converging keywords*. A detailed description of the evaluation methodology is also presented in this chapter following both quantitative and qualitative analyses.

Chapter 6: Leap2Trend: Detecting The Contextualising Keywords. This chapter describes the third contribution of this thesis. It introduces *Leap2Trend*, a novel approach to early detection of research trends that relies on temporal word embeddings to track the dynamics of similarities between pairs of keywords, their rankings and respective uprankings (ascents) over time. This approach also represents the first path that this thesis follows to perform the computational history of science by detecting the emerging scientific trends in the area of machine learning, and it is devoted to the second definition given to the emerging scientific trends – *contextualising keywords*. Furthermore, this chapter

details the rigorous evaluation method that is proposed and it relies on solid standards such as Google Trends hits and Google Scholar citations to validate the proposed approach in two datasets related to the area of machine learning and bioinformatics.

Chapter 7: Vec2Dynamics: Tracking The Dynamism of Keywords. This chapter describes the fourth contribution of this thesis. It introduces *Vec2Dynamics*, a temporal word embedding approach that reports the stability of k-nearest neighbors of scientific keywords over time to check whether they are taking new neighborhood due to evolution of scientific literature, and accordingly track their dynamics over time. This approach represents the second path that this thesis follows to perform the computational history of science by tracking the dynamism of scientific keywords over time. In this chapter, a descriptive analysis is performed to verify the efficacy of the proposed approach in the area of machine learning.

Chapter 8: Conclusion and Future Directions. This chapter summarises the thesis contributions and results. Moreover, a discussion about current and future challenges of scholarly data mining and trend analysis is provided.

Chapter 1	**Introduction**	
Chapter 2	**Scholarly Data Mining**	**Literature review (Dridi et al., 2021)**
Chapter 3	**Word Embedding Techniques – Word2vec**	**Background**
Chapter 4	**k-NN Embedding Stability for Word2vec Hyper-parametrisation**	**Contribution 1 (Dridi et al., 2018)**
Chapter 5	**Hist2Vec: Detecting The Converging Keywords**	**Contribution 2 (Dridi et al., 2019a)**
Chapter 6	**Leap2Trend: Detecting The Contextualising Keywords**	**Contribution 3 (Dridi et al., 2019b)**
Chapter 7	**Vec2Dynamics: Tracking The Dynamism of Scientific Keywords**	**Contribution 4 (under review)**
Chapter 8	**Conclusion and Future Directions**	

TABLE 1.1: Thesis outline

Chapter 2

Scholarly Data Mining

> "Knowledge is but a body, of which intelligence is the soul."
>
> — Walter Moxon, M.D., F.R.C.P. (1836–1886)

After stating the research problem addressed in this thesis and introducing the main research area related to the contributions – which is the area of *scholarly data mining* – in the previous chapter, this chapter presents the state-of-art on scholarly data mining. In particular, it initially considers the research interest analysis of scholarly data in Section 2.1. This is followed by a survey on the applications of scholarly data mining and the analysis methods used in these applications in Sections 2.2 and 2.3, respectively. It then shows the areas of application of scholarly data mining in Section 2.4. Afterwards, the chapter discusses the open challenges in the area of scholarly data mining, and how this thesis contributes to address some of these challenges in Section 2.5. Finally, the chapter is concluded by a summary in Section 2.6. The work described in this chapter has been published in (Dridi et al., 2021).

2.1 Research Interest Analysis

Although the study of scholarly data is relatively new, some studies have emerged (Feng et al., 2017) on how to investigate scholarly data usage in different disciplines. These studies motivate investigating the scholarly data generated via academic technologies such as scholarly networks and digital libraries for building scalable approaches for retrieving, recommending and analysing the scholarly content. Consequently, this has spawned five key applications that are *citation analysis* (Dey et al., 2017; Shi et al., 2015; Zehra and Umut, 2018; Ying et al., 2014), *document analysis* (Cornelia et al., 2015; Shardlow et al., 2018; Tuarob et al., 2016; S. Kim et al., 2018), *conference analysis* (Effendy et al., 2014; Effendy and Roland, 2016), *trend analysis* (An et al., 2017; Dridi et al., 2019b; Hou et al., 2018; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017), and *literature analysis* (Y. Liu et al., 2015; Tan et al., 2016; Tang, 2016; Tang et al., 2008; H. Li et al., 2006; Dunne et al., 2012; Osborne et al., 2013). Nevertheless, due to the increasing interest to scholarly data mining, it becomes essential to closely study the approaches for scholarly data analysis, categorise them based on the literature features or explore the techniques involved in mining scholarly data. In this regards, the aim of this chapter is to systematically review the most interesting research works published on the use of scholarly data mining. An important collection of research articles has been analysed with special

attention paid to the investigated literature features and the different analysis methods used. The final aim is therefore to provide the readers with a systematic revision about existing scholarly data mining applications: the involved techniques and the application areas.

2.1.1 Research Motivation

The importance of *scholarly data mining* has been raised for several key reasons. First, it is the availability of abundant academic resources. In addition to the scholarly documents such as papers, books, reports, *etc*, multiple associated data are available today including information about authors, citations, institutions, funds and academic networks (J. Liu et al., 2018). Furthermore, there have been several initiatives by governments and organisations to digitise academic resources in order to meet the challenges of information explosion. As a matter of fact, the scholarly communication has been revolutionised drastically over the past two decades due to the unprecedented advancement in information and communication technology. This latter has brought a revolutionary form in archiving and accessing knowledge in the digitised form that used to be in the conventional print form. Second, due to this easily accessible reservoir of data, researchers and scholars are in an elevating need for a deeper understanding of the structure and dynamics of science. In other words, sophisticated techniques and tools are highly solicited to help researchers to learn better about knowledge production processes, curate insights from scholarly data and speculate upcoming research topics. Third, the availability of this vast amount of data about scientists' collaborations, document sharing and publications enables the evaluation of scientific impact of different entities including papers, authors and journals. The measurement of this scientific impact is deemed vital for governments and businesses for decision-making processes such as funding allocation, research gap identification, university ranking determination, tenure and recruitment decisions. Finally, going beyond the study of the scientific impact, scholarly data analysis also promotes the understanding of human social activities. It provides sociologists with valuable data to observe researcher interactions and community formation. It also allows countries to evaluate the impact of institutions or scientists to allocate resources. Overall, scholarly data mining contributes to the *Science of Science* (Fortunato et al., 2018; Light et al., 2014) that advances the understanding of the structure and dynamics of science.

According to the literature, there have not been enough attempts to closely study the approaches of scholarly data mining, categorise them based on the literature features or explore the techniques involved in mining scholarly data. An effort in this direction could reveal new branches for future research in this area. In fact, there have been some recent reviews related to scholarly data but not deeply exploring the aforementioned specific issue. For instance, on the one hand, two reviews (Khan et al., 2017; Feng et al., 2017) considered scholarly data from a general perspective. The authors studied the use of big data in scholarly ecosystems, starting from scholarly data management and relevant technologies, through data analysis methods, to finally looking into the research issues. On the other hand, another three reviews were specific and had a narrow perspective. The first of these reviews (Ying et al., 2014) provides a comprehensive overview of citation analysis in terms of its theoretical foundations, methodical approaches and example applications. The second review (J. Liu et al., 2018) addressed the issue of scholarly data visualisation by focusing on the visualisation tools and analytic systems. The third review (Bai et al.,

2019) dealt with the scientific recommendation problem as a sub-problem of scholarly data analysis. It provided a comprehensive review on the scholarly paper recommendation by reviewing the recommendation algorithms, introducing the evaluation methods of different recommender systems and highlighting the open issues in the paper recommendation systems.

These reviews are all recent, which shows a great interest in the topic of scholarly data mining, not only for the opportunities it offers to the scientists and scholars to understand the unprecedented amount of scholarly data freely available, but also for institutions and governments that could take benefit from the proposed approaches for decision-making processes.

2.1.2 Research Process

In the search process, two methods have been followed to obtain the articles reviewed in this chapter:

- **Database Search** using queries related to the research area of this thesis. The following databases were used to gather articles combining scholarly data mining tools, techniques and applications: Web of Science [1], Google Scholar [2], DBLP [3], ScienceDirect [4], ACM Digital Library [5], and IEEE Xplore Digital Library [6]. In each of these databases, the used search queries search for terms – related to the topic of the review – in the title of publications, such as *"scholarly data mining"*, *"knowledge discovery"*, *"scientific data"*, *"trend analysis"*, *"scientific recommendation"*, *"citation analysis"*, and *"bibliometrics"*.

- **Selection of related publication venues** that are known to publish in the area of scholarly data and related topics. Specifically, two venues are considered. The first venue is Scientiometrics journal [7], which is a peer reviewed journal concerned with the quantitative aspects of the Science of Science and scientific research. The second venue is the Joint Conference on Digital Libraries (JCDL) [8] that represents a major international forum focusing on digital libraries and associated issues.

As the search queries used were a mixed bag of generic and specific terms, the results were refined by reading the titles and the abstracts within the articles, filtering out those that did not align with the scope of this review. Particularly, all papers had to be about applications of scholarly data mining and knowledge discovery from scientific data. All the works studied in this thesis were published in peer reviewed journals or top conferences, which guarantees that they satisfy a certain standard of quality.

[1] https://login.webofknowledge.com/
[2] https://scholar.google.com/
[3] https://dblp.uni-trier.de/
[4] https://www.sciencedirect.com/
[5] https://dl.acm.org/
[6] https://ieeexplore.ieee.org/Xplore/
[7] https://link.springer.com/journal/11192
[8] https://www.jcdl.org/

2.1.3 Research Statistics

An important number of publications has been analysed in this thesis. The increasing interest in the research of scholarly data mining becomes clear from the yearly increase in the number of publications related to this area mainly in the last five years starting from 2014 (see Figure 2.1 that shows the distribution of the number of publications studied in this chapter until 2019, which is the last year of the studied literature, marked in green color. The Y-axis represents the raw number of publications per year). However, it is worth mentioning that the field traces its roots back in the late 1980's and mid 1990's through initial empirical studies principally on citation analysis. These early studies were focusing on the understanding of the citation behavior of scientific community. This may have contributed to the study of only citations as a literature feature of scientific publications during the first decade that preceded the emergence of the field. Also, this may justify why these analyses were mainly tackled in the area of information science rather than the area of computer science.

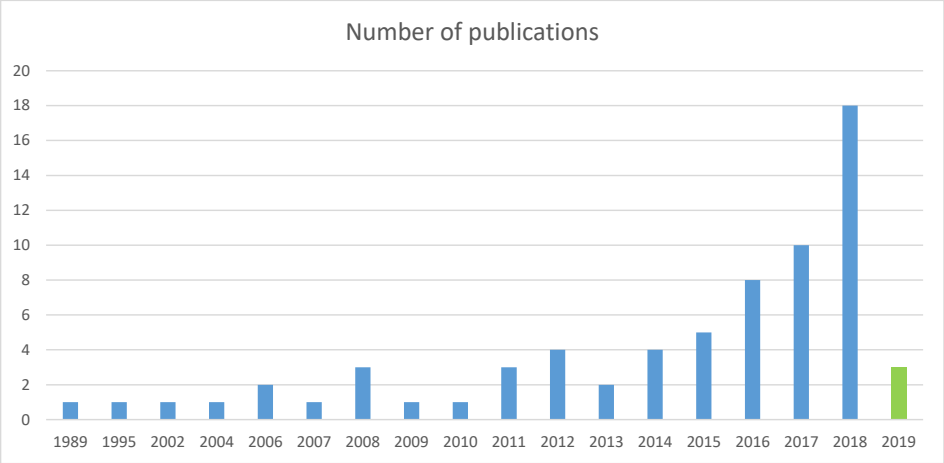


FIGURE 2.1: Number of publications by year

It is also interesting to analyse the geographic distribution of publications. To this end, the country of publications' authors is considered. Figure 2.2 shows the total number of publications corresponding to each country. Recall that the data collected corresponds to the literature analysed in this chapter. What can be clearly seen in this figure is the dominance of both USA and China in publishing in the area, followed by the UK and Germany. Interestingly, this highly correlates with the report on the research outputs by country[9], published by *Nature Index* in 2019. Also,

[9] https://www.natureindex.com/country-outputs/generate/All/global/All/n_article

it aligns with the global Research and Development (R&D) spending by country according to the latest statistics from the UNESCO Institute for Statistics[10].

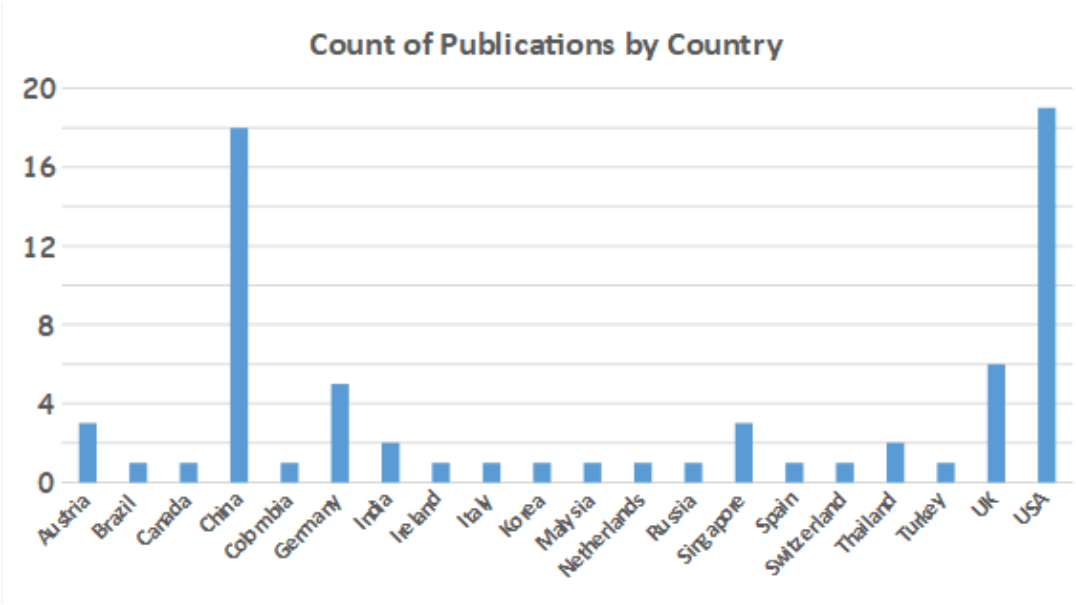


FIGURE 2.2: Number of publications by country

A further interest indicator is the number of citations of each work. The mean number of citations per year in the last decade is analysed. The results are very sensitive to the low number of works per year that is in the pool of studied articles. Table 2.1 illustrates the mean number of citations. It is somewhat intuitive that the mean number of citations of the papers published early in the time is significantly higher than the one of those published recently. However, it is worth noting that the mean number of citations is also relatively high during the last two years of the analysis (2018-2019). This is insightful as it shows that the research area of scholarly data mining is developing and becoming of interest nowadays.

2011	2012	2013	2014	2015	2016	2017	2018	2019
29.33	93	61.5	66.75	13.6	15.87	13.1	7.4	3.33

TABLE 2.1: Mean number of citations of the papers published between 2011 to 2019 by year

2.2 Literature-based Analysis

The ultimate goal of scholarly data mining is to understand the relational structure of science and provide the scholars with better academic services such as academic recommendation and literature organisation. To this end, scholarly data mining involves various applications, which are mainly categorised based on the literature features: documents, citations, conferences and trends. In this section, these applications are examined with respect to the aforementioned literature features. Table 2.2 summarises these applications and categorises the references with respect to the literature features they have investigated.

[10] http://uis.unesco.org/en/news/new-uis-data-sdg-9-5-research-and-development

2.2.1 Citation Analysis

As citation counts remain the most important measure of scientific impact, citations – as a literature feature – have presided the attention of researchers investigating the area of scholarly data mining. Therefore, citation analysis has been widely explored as an application of scholarly data mining.

Citation analysis has been extensively used to understand the scholarly communication through citation patterns (Ying et al., 2014). It does not only aim to assess the impact of research outputs, but it also aims to reveal the scholarly communication, map the landscape of scientific disciplines and track the knowledge transfer across domains.

In addition to the conventional study of citation and co-citation network to measure the impact of scientific papers, citation analysis involves different patterns that include *readership counts* (Aduku et al., 2017; Maflahi and Thelwall, 2018; Thelwall, 2018), *bibliometrics* (Godin, 2006; Lv et al., 2011; Martínez-Gómez, 2015; McBurney and Novak, 2002; Monroy and Diaz, 2018; Pilkington, 2004) and even the metrics derived from social media – termed as *altmetrics* (Bornmann and Haunschild, 2018; Nabout et al., 2018; Priem and Costello, 2010; Weller et al., 2011).

Citation and Co-citation Analysis

Citation and co-citation analysis is shaped around citation counting and citation relationships between documents/scholars that are cited together by other documents/scholars (Dey et al., 2017; Shi et al., 2015; Zehra and Umut, 2018).

Citation analysis mainly studies the citation network, where nodes represent papers, authors, or journals, and edges represent the number of times each paper/author has been cited, co-authored, or co-cited (Ying et al., 2014). It then measures the impact of published research quantitatively, and could be termed as *count-based citation analysis*. While citation count remains essential to measure the scientific impact, it fails to address the "how and why" questions of citation analysis. To fill this gap, *content-based citation analysis* (Ying et al., 2014) has been proposed as the next generation of citation analysis. It aims to study both syntactic and semantic levels of citations. The syntactic level considers the location of the reference in a citing article while the semantic level studies why a reference has been cited in a citing article. The content analysis of citations has included both manual approach and semi-automatic approach of natural language processing (NLP). The main goal of content citation analysis is then to develop a code-book used to annotate citation contexts. NLP techniques have been used to extract the key concepts from citation contexts to understand the citing behavior. However, the identification of the best window size to extract the proper citation context and the detection of the correct citing paper sections are still an open challenge.

Co-citation analysis studies citation relationships in the co-citation network (Star Zhao and Ye, 2013). It measures the frequency with two papers/scholars are cited together by other papers/scholars. Typically, there are mainly two types of co-citation analysis methods namely *author co-citation analysis* (Jeong et al., 2014) and *document co-citation analysis* (Trujillo and Long, 2018). The author co-citation analysis measures the similarity between co-cited authors by considering author's citation context. To this end, the citing sentences are extracted to obtain the topical relatedness between the cited authors instead of traditional author co-citation frequency. The citing sentence similarity is then measured by topical relatedness between two

citing sentences. However, the document co-citation analysis enables to identify relevant literature and scholarly communities that may be left unnoticed in standard approaches to literature searching. Resulting networks help to identify gaps between published research areas. Document co-citation analysis is then proposed as a potential methodology to promote trans-disciplinary. In (Trujillo and Long, 2018), the authors have explored 229 source articles from the literature of systems thinking, extracted from the Web of Science Core Collection[11]. After generating the document co-citation network, the authors have explored patterns in influential literature developed across different disciplines. For instance, they have demonstrated that community structure could be detected within the co-citation networks for systems thinking. Both of them enable the identification of the intellectual structure of a research domain and the recognition of relevant scholarly communities.

Both citation and co-citation analyses have been used for different scholarly aims such as author names ambiguity (Sun et al., 2011), topic classification (Cornelia et al., 2015), scientific success prediction (Acuna et al., 2012), identification of sleeping beauties (Dey et al., 2017), identification of dynamic knowledge flow patterns (An et al., 2017) and trend analysis (Hou et al., 2018).

Readership Analysis

Readership analysis studies *Mendeley*[12] reader counts – that correspond to the number of readers of each article – and their evidence of early scholarly impact for published articles. Different works (Aduku et al., 2017; Maflahi and Thelwall, 2018; Thelwall, 2018) have studied whether Mendeley reader counts reflect the scholarly impact of publications. While (Maflahi and Thelwall, 2018; Thelwall, 2018) have focused their studies on investigating the early scholarly impact of Mendeley reader counts for journal articles, (Aduku et al., 2017) have studied whether this impact is equally true for conference papers. To do so, the authors have extracted Mendeley readership data and Scopus[13] citation counts for both journal articles and conference papers published in 2011 in computer science and engineering. The authors have found Mendeley a moderate correlation between readership counts and citation counts for both journal articles and conference papers in computer science. However, the correlations were much lower between Mendeley readers and citation counts for conference papers than for journal articles in engineering. Therefore, there seem to be disciplinary differences in the usefulness of Mendeley readership counts as impact indicator for conference papers. Overall, all research works investigating Mendeley reader counts have found significant positive correlations between readership counts and citation counts, while Mendeley reader counts appear before citations. Readership analysis has been then proposed as a valuable early impact indicator for published research, addressing the issue of citations that take time to accumulate.

Bibliometrics Analysis

Bibliometrics analysis (Monroy and Diaz, 2018) focuses on the use of statistical analysis to examine scientific production patterns in a scientific field (Godin, 2006; McBurney and Novak, 2002). For instance, the authors in (Monroy and Diaz, 2018)

[11] https://clarivate.com/webofsciencegroup/solutions/web-of-science-core-collection/
[12] https://www.mendeley.com/
[13] https://www.scopus.com/home.uri

have applied time series tools to bibliometric data to conduct a comparative study of the dynamics of scientific production for several countries, in terms of papers published. They have compared the histories of scientific development of countries, aiming to understand the causes and circumstances that led to dynamics of knowledge production. They have then identified the dynamical changes that affected global scientific production, and the instances, where global production was influenced by social, political and economic circumstances. On the other hand, bibliometrics have applied statistical analysis to assess relationships between authors, entities, journals or countries, in addition to measuring the impact of research and linkage involving co-citations and keywords employed (Lv et al., 2011; Martínez-Gómez, 2015; Pilkington, 2004). In this context, the authors in (Lv et al., 2011) have applied statistical analysis and knowledge visualisation technology to study graphene literature from different subjects, authors, countries and keywords distributed in several aspects of research topics. For this matter, the authors have collected and analysed data from 1991 to 2010 from the *Science Citation Index* database, *Conference Proceeding Citation Index* database and *Derwent Innovation Index* database integrated by Thomson Reuters[14]. Their bibliometric analysis has shown that the clusters distributed regularly in keywords of applied patents in recent 5 years due to the potential applications of graphene research gradually found.

Bibliometrics analysis is centrally, but not only, based on citation analysis. It also involves descriptive linguistics (Gleason, 1961), the development of thesauri, the evaluation of reader usage, and the analysis of associated keywords. All these bibliometrics patterns are used to identify research clusters, emerging topics and leading scholars in bitcoin literature by analysing 1162 papers indexed in Web of Science (Merediz and Aurelio, 2019).

Bibliometrics are frequently used in the field of library and information science. A sub-filed of it – that is concerned with the study of scientific publication – is called *scientometrics*, which is defined as "the study of the quantitative aspects of the process of science as a communication system" (Mingers and Leydesdorff, 2015).

Altmetrics Analysis

Altmetrics analysis (Bornmann and Haunschild, 2018; Nabout et al., 2018; Priem and Costello, 2010; Weller et al., 2011) supports the use of activities on online social media platforms as an early signal of research impact for scientific publications. Altmetrics seek new means of quantifying the impact of research outside the realm of research papers, such as online media and social network. This class of metrics includes mentions in the news, blogs, and on Twitter[15]; article page-views and downloads; GitHub[16] repository watchers.

Altmetrics have been considered as a measure of scientific dissemination and an early indicator of scientific influence and impact. For instance, they can point to interesting spikes in different types of attention. As a proof of evidence, some studies (Nabout et al., 2018) have proven that altmetrics are concordant with citation-based metrics. By way of illustration, (Nabout et al., 2018) have studied the correlation between traditional citation-based indicators and activities on online social media platforms in a dataset of 2,863 papers published in five ecological journals. Their results supported the use of activities on online social platforms as an early

[14] https://www.thomsonreuters.com/en.html
[15] https://twitter.com/
[16] https://github.com/

signal of research impact of ecological articles. However, this outcome is not totally supported by (Bornmann and Haunschild, 2018) who studied Twitter dataset to measure the impact of science and found that without considering the content of the tweets, simple counting can lead to wrong conclusions.

2.2.2 Document Analysis

A document represents an important literature feature in scholarly communication, which is defined by a set features itself including *author*, *content*, *structure*; and it is used for different scholarly usages such as recommendation and summarisation. Document analysis includes then subsequent analyses detailed below.

Authorship Analysis

Authorship analysis has been treated differently in the literature. For instance, the authors in (Rexha et al., 2018) have proposed to associate segments of text with their real authors using content-agnostic and stylometric features to solve the problem of authorship identification. For this purpose, two pilot studies have been conducted on a selected data from the free database created by the US National Library of Medicine – PubMed[17]. Both studies aimed to understand how humans can identify authorship among documents with high content similarity. The first study was a quantitative experiment involving crowd-sourcing, while the second was a qualitative one executed by the authors. Both experiments and observations contribute to automating the process of authorship identification as well as to distinguish specific features used by humans in their decision-making process. In (Sun et al., 2011), however, the authors have explored heuristic features based on citations and crowd-sourced topics to detect ambiguous author names in the context of social citation analysis systems such as *Scholarometer*[18] system. Two classes of features were used. The first is a heuristic based on the percentage of citation accrued by the top name variations for an author, while the second feature class relies on crowd-sourced data to detect ambiguity at the topic level. The proposed approach succeeded to detect ambiguous author names in crowd-sourced scholarly data with an accuracy of 75%.

Document Structure Analysis

Document structure analysis (Boyack et al., 2018; Heffernan and Teufel, 2018a; Lu et al., 2018a) studies the internal document structure by identifying the functional structure (further detailed at three levels: section header-based identification, section content-based identification or paragraph-based identification (Lu et al., 2018a)), identification of problems and solutions in a specific paper by making a binary decision about problem-hood and solution-hood of a given phrase in the paper (Heffernan and Teufel, 2018a), or by studying research proposals and analysing their discourse for clarity (Boyack et al., 2018). For the functional structure identification, the authors in (Lu et al., 2018a) have proposed a novel clustering algorithm to generate a domain-specific functional structure, applied to 300 research papers in computer science. The application of the proposed approach, in two tasks: academic search and keyword extraction, confirms that the identified structure obtains more

[17] https://pubmed.ncbi.nlm.nih.gov/
[18] https://scholarometer.indiana.edu/

relevant information and achieves better performance. However, for the identification of problems and solutions, the authors in (Heffernan and Teufel, 2018a) have proposed an automatic classifier that makes a binary decision about problem-hood and solution-hood of a given phrase, that may or may not be a description of a scientific problem or a solution. The authors have defined a set of 15 features, including syntactic information (part-of-speech (POS) tags), document and word embeddings, and have applied several machine learning algorithms such as Naïve Bayes, Logistic Regression and Support Vector Machine, on a corpus of 2000 positive and negative examples of problems and solutions extracted from the 2016 Association of Computational Linguistics (ACL) anthology. The obtained results reveal the ability of the proposed classifier to distinguish problems from non-problems with an accuracy of 82%, and solutions from non-solutions with an accuracy of 79%. Regarding research proposal analysis, the authors in (Boyack et al., 2018) have used both citation and discourse analyses of 369,501 proposals submitted to the U.S. National Institutes of Health (NIH) by the University of Michigan Medical School, to discover possible predictors of proposal success. The analyses have focused on two issues: the Matthew effect in science – Merton's claim that eminent scientists have an inherent advantage in the competition for funds – and quality of writing or clarity. The obtained results suggested that a clearly articulated proposal is more likely to be funded than a proposal with lower quality of discourse.

Document structure analysis also includes the study of slide presentations (Weber and Gunawardena, 2011). It investigates the use of knowledge units – that represent scientific knowledge by combining elements of procedural, declarative, and structural knowledge – for the automated construction of slides. The knowledge units have been defined as the three paradigms of the research process background, progress and completed.

Content Analysis

Document content analysis treats the document in two ways: a coarse-grained way by studying only keywords (S. Kim et al., 2018) and a fine-grained way by digging into the paper textual content (Cornelia et al., 2015; Shardlow et al., 2018; Tuarob et al., 2016).

Keyword analysis (S. Kim et al., 2018) examines connections between keywords used to describe theses and dissertations in order to vividly picture similarities and differences among research domains. In this context, the authors in (S. Kim et al., 2018) have analysed data from 29,435 dissertations and theses found in the ProQuest Theses and Dissertation database[19] in the years 2009–2014. The obtained results identified interdisciplinary clusters, as well as the key differences in connections between the four computing disciplines in the database: computer science, computer engineering, information technology and information science. However, textual content analysis involves different applications such as topic classification (Cornelia et al., 2015), identification of research hypotheses (Shardlow et al., 2018) and extraction of algorithms (Tuarob et al., 2016). For topic classification, the authors in (Cornelia et al., 2015) have proposed a co-training approach that uses the text and citation information of a research article as two different views to identify the topic of an article. A subset sampled from the *CiteSeer*^{x} digital library[20] – consisting of

[19] https://about.proquest.com/products-services/databases/

[20] https://citeseerx.ist.psu.edu/

3,186 labeled papers – has been used for topic classification with a co-training classifier. The obtained results showed that the proposed approach performed better than other semi-supervised and supervised methods. However, for the identification of research hypotheses, the authors in (Shardlow et al., 2018) have proposed a supervised method to extract new meta-knowledge dimensions that encode research hypotheses. A corpus of one thousand MEDLINE[21] abstracts on the subject of transcription factors in human blood cells has been used, and a random forest classifier has been applied to achieve a better performance than previous efforts in detecting knowledge type, with a precision ranging from 86% to 100%. Regarding the extraction of algorithms, the authors in (Tuarob et al., 2016) have developed *AlgorithmSeer*, a system for extracting and searching for algorithms. To do so, hybrid machine learning approaches have been proposed to discover algorithm representations, and different techniques have been adopted to extract textual metadata for each algorithm. Finally, a demonstration version of *AlgorithmSeer* that is built on Solr/Lucene open source indexing and search system is presented and applied to over 200k algorithms extracted from over 2 million scholarly documents.

Scientific Recommendation

Scientific recommendation includes topic recommendation (Alam and Ismail, 2017) and reviewer recommendation (Shu Zhao et al., 2018a). Scientific paper recommendation has been provided to assist scholars in finding relevant papers across the tremendous amount of academic information in the era of big scholarly data. In this context, (Kong et al., 2018) have developed *VOPRec*, a scientific paper recommendation system based on vector representation learning of paper in citation networks. In fact, paper recommendation takes into account both text information of papers and structural identity with the citation network. Similarly, topic recommendation hinges upon bibliometric information of the literature to identify a suitable topic of current importance from a plethora of research topics. In this context, (Alam and Ismail, 2017) have developed *RTRS* – a recommender system for academic researchers – to assist both novice and experienced researchers in selecting research topics in their chosen field.

Scientific Text Summarisation

Scientific text summarisation has been proposed to help scholars to know about the most influential content of the paper due to the vast growth of literature that makes it difficult for them to find high impact papers on unfamiliar topics (Mei and Zhai, 2008). Different approaches have been proposed to generate summaries of scientific papers. The authors in (Rahul et al., 2013) have presented a system that takes a topic query as input and generates a survey of the topic by first selecting a set of relevant documents, and then selecting relevant sentences from those documents. In (Qazvinian and Radev, 2008), on the other hand, the authors have proposed a citation summary network that uses a clustering approach, where communities in the citation summary's lexical network are formed and sentences are extracted from separate clusters. Another summarising problem has been tackled by (Mei and Zhai, 2008), which is summarizing the impact of a scientific publication. The authors have used language modeling methods – that incorporate features such as authority and

[21] https://www.medline.com/

proximity extracted from the citation context – to extract sentences that can represent the most influential content of the paper. The scientific summarisation includes also summarising document-elements like tables, figures, and algorithms in scientific publications to augment search results and enable the retrieval of these document-elements (Bhatia and Mitra, 2012).

2.2.3 Conference Analysis

Conference analysis (Effendy et al., 2014; Effendy and Roland, 2016) studies conference categorisation (Effendy and Roland, 2016) and relatedness between conferences (Effendy et al., 2014). A case-study approach was adopted by (Effendy et al., 2014) to assess the relatedness measures between conferences in computer science based on the computer science bibliography DBLP[22]. They have shown that the relatedness ranking produced can correlate well with the reputation ranking from the Australian Computer Research and Education conference ranking (CORE)[23].

Both studies help to understand the basis of conference reputation ratings, determine what conferences are related to an area and the classification of conferences into areas.

2.2.4 Trend Analysis

Trend analysis has received considerable interest in the past few years, because finding a research trend is a key to find a niche in a particular field of interest, especially for those new to this field. The main goal of trend analysis is to reveal hidden trends within these vast resources, such as research trend evolution and community dynamics (Feng et al., 2017).

Different approaches in the literature dealt with trend analysis using different features such as citation counts, paper content especially keywords, or both of them. These approaches can the be categorised into three categories with respect to the features they have been using: (i) *bibliometrics-based approaches* (An et al., 2017; Hou et al., 2018; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017) that are based on social network analysis, citation and co-citation analysis, (ii) *content-based approaches* (Dridi et al., 2019a; Dridi et al., 2019b; Weismayer and Pezenka, 2017) that treat entities – essentially keywords – reflecting the paper content (Weismayer and Pezenka, 2017) or dig deeply into the paper content and study the associations between keywords (Dridi et al., 2019a; Dridi et al., 2019b) and (iii) *hybrid approaches* (Effendy and Yap, 2017; Hoonlor et al., 2013) that combine both citation and content.

The bibliometrics-based approaches rely mainly on citation counts of published papers, and consequently find clues to topic evolvement (Zehra and Umut, 2018). For instance, the authors in (An et al., 2017) have considered both backward and forward citations to propose a hidden markov model to identify temporal patterns of knowledge flows in business method patents. However, (Hou et al., 2018) have used a document co-citation analysis of a subsequent 7,574 articles published in 10 information science (IS) journals between 2009 and 2016, including 20,960 references, to study changes in the research topics in the IS domain. Similarly, (Rossetto et al., 2018) have used citation and co-citation analysis to understand what are the main theoretical pillars that support the structure of innovation theories and fields. While citation counts may infer the importance of scientific work, they fail to delve

[22] https://dblp.uni-trier.de/
[23] https://www.core.edu.au/

into the paper content, which could lead to a more accurate computational history. For this reason, content-based approaches have emerged. For instance, some emerging works (Ashton et al., 2012; Hall et al., 2008; Mortenson and Vidgen, 2016) have proposed topic models to study the dynamics of research topics and accordingly the progress of science. While topic models try to extract semantics by capturing document level associations between words, they fail to detect pairwise associations between keywords. To overcome this problem, word embedding techniques have been proposed to conduct a fine-grained content analysis of scientific content. In this matter, (J. He and Chen, 2018) have proposed the first work that aimed to track the semantic changes of scientific terms over time in the biomedical area. The other work in this direction has been proposed by this thesis (Dridi et al., 2019a; Dridi et al., 2019b) that introduced a temporal word embedding approach for computational history applied to machine learning publications. The approaches detect the converging keywords that may result in trending keywords by computing the acceleration of similarities between keywords, their rankings and uprankings over successive timespans.

The hybrid approaches use both citation analysis and content analysis to detect research trends. For instance, Hoonlor et *al.* (Hoonlor et al., 2013) have analysed data on grant proposals, ACM[24] and IEEE[25] publications using sequence mining, bursty word detection and clustering. In like manner, Hou et *al.* (Hou et al., 2018) tracked the evolution of research topics between 2009 and 2016 using the timeline knowledge map through Document-Citation Analysis (DCA) of articles published in information science journals. They employed dual-map overlays of the information science literature to trace the evolution of the knowledge base of IS research based on scientometric indicators (H-index), citation analysis and scientific collaboration. In the same context, Effendy and Yap (Effendy and Yap, 2017) obtained the computational history using the Microsoft Academic Graph (MAG)[26] dataset. In addition to the citation-basic method, they used a content-based method by leveraging the hierarchical *Field of Study (FoS)* given by MAG for each paper to determine the level of interest in any particular research area or topic, and accordingly general publication trends, growth of research areas and the relationship among research areas in computer science.

These approaches have been applied to a wide range of disciplines such as relations and economy (Soriano et al., 2018), innovation and entrepreneurial ecosystem (C. Zhang and Guan, 2017), business (Rossetto et al., 2018) and business model innovation (An et al., 2017), marketing and tourism (Weismayer and Pezenka, 2017), medical domain (Boyack et al., 2018), biology (Y. Liu et al., 2015), information science (Hou et al., 2018) and computer science (Alam and Ismail, 2017; Dey et al., 2017; Effendy et al., 2014; Effendy and Yap, 2017; Effendy and Roland, 2016; Hoonlor et al., 2013; S. Kim et al., 2018).

This thesis concentrates on the areas of computer science (CS) – namely machine learning – and bioinformatics. While no previous work on predicting research trends in bioinformatics is present, research findings on trend analysis within computer science is reported in the following. For instance, Hoonlor *et al.* (Hoonlor et al., 2013) were the early researchers interested in learning about the evolution of CS research. They analysed data from 1990 to 2010 on proposals for grants supported by the U.S

[24] https://dl.acm.org/

[25] https://ieeexplore.ieee.org/

[26] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

National Foundation[27] and on CS publications in the ACM Digital Library[28] and IEEE Xplore Digital Library[29] using sequence mining, bursty word detection and clustering, network extraction and visualisation. They aimed to investigate changes over time in the CS research landscape; interaction of CS research communities; similarities and dissimilarities between research topics. Similarly, Hou *et al.* (Hou et al., 2018) revealed the evolution of research topics between 2009 and 2016 using the timeline knowledge map through Document-Citation Analysis (DCA) of 7,574 articles published in 10 information science (IS) journals including 20,960 references. They used dual-map overlays of the IS literature to track the evolution of the knowledge base of IS research based on scientometric indicators (H-index), citation analysis and scientific collaboration. In the same context, Effendy and Yap (Effendy and Yap, 2017) performed trend analysis using the Microsoft Academic Graph (MAG)[30] dataset. But, in addition to the bibliometric-basic method (citation analysis), they applied a content-based method by using the hierarchical *FoS (Field of Study)* provided by MAG for each paper to measure the level of interest in any particular research area or topic, and consequently revealed general publication trends, evolution of research areas and the relationship among research areas in CS.

Both approaches described above can be categorised as hybrid approaches. They combine the citation analysis with the content analysis to reveal research trends. The content analysis only studies bursty keywords in (Hoonlor et al., 2013) and fields of studies in (Effendy and Yap, 2017) without drilling into the paper content or following a fine-grained analysis. Instead, they focused on citation analysis to reveal citation trends and consequently reveal the evolution of research areas. While citation counts are deemed essential to evaluating the importance of scientific work, the citing behavior could possibly be for non-scientific reasons (Bornmann and Daniel, 2008). Moreover, citations can take months to even years to stabilise enough to reveal research trends. As a matter of fact, there can be interesting papers – termed as *sleeping beauties* (Dey et al., 2017) – which do not get cited much for several years after publication, but then unexpectedly start getting cited.

For these reasons and the fact that citation-based approaches fail to dig into the paper content, the work presented in this thesis tends to be placed in the category of content-based approaches by following a fine-grained content analysis of research papers.

In this direction, some work has begun to emerge. Anderson *et al.* (Ashton et al., 2012) have developed a people-centric methodology for computational history that tracks the flow of authors across topics to discern how some sub-fields flow into the next, forming new research directions. This methodology is based on a central phase of topic modelling that classifies papers into topics and identifies the topics the authors contribute to. In the same context, Salatino *et al.* (Salatino et al., 2017) have proposed *Augur*, which is an approach that analyses the diachronic relationships between research areas and detects clusters of topics that exhibit dynamics of already established topics. Similarly, Li *et al.* (Li et al., 2019) have recently proposed an improved method by introducing WordNet to LDA in order to find latent topics of large corpora of NIPS publications and discover the dynamics of research topics. To do so, their method groups the documents by time in each topic. Then, it

[27] https://www.nsf.gov/
[28] https://dl.acm.org/
[29] https://ieeexplore.ieee.org/
[30] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

counts the number of documents by time, which helps to reveal whether the topics are rising or falling in popularity.

While these approaches (Ashton et al., 2012; Li et al., 2019; Salatino et al., 2017) intend to perform a content analysis of research papers by applying topic modelling, they still suffer from the delay in time in the detection of trends. For instance, both the flow of authors across topics and the dynamics of established topics take time to happen. In addition, topic modelling – as a natural language processing technique used for these content-based approaches – is not able to detect pairwise associations between words while the study of these associations could lead to the detection of emerging trends at a very early stage.

2.2.5 Literature Analysis

Literature analysis encloses more than one literature feature (H. Li et al., 2006; Dunne et al., 2012; Y. Liu et al., 2015; Osborne et al., 2013; Tan et al., 2016; Tang, 2016; Tang et al., 2008; Tao et al., 2017). It studies the key nodes of the academic social network such as papers, authors, citations and corresponding relationships at the same time.

Generally, literature analysis has been shaped around the development of new tools and systems that support the exploration of scholarly data. This has been seen in the case of development of academic search systems that aim to comprehensively search and mine literature (Y. Liu et al., 2015; Tan et al., 2016; Tang, 2016; Tang et al., 2008) such as *ArnetMiner* (Tang et al., 2008), *AMiner* (Tang, 2016) and *CiteSeerX* (H. Li et al., 2006). Another example of these tools is the study maps that have been built efficiently and thoroughly through topic analysis methods to dig into the underlying principles of a specific paper (Tao et al., 2017). Also, visualisation has gained a great interest in literature analysis approaches because it helps to describe, analyse, simulate an academic social network and support community detection and collaboration networks. For instance, *Action Science Explorer* (ASE) (Dunne et al., 2012) has been developed to show citation patterns and identify clusters; and *Rexplore* (Osborne et al., 2013) has integrated statistical analysis, semantic technologies and visual analytics to provide effective support for exploring and making sense of scholarly data.

2.3 Scholarly Data Mining Methods

Scholarly data mining has been realised with different methods including statistical and empirical analysis, social network analysis, machine learning techniques, and natural language processing techniques. In the following, the methods of scholarly data mining are briefly introduced and the applications they have been used for are specified. Table 2.3 summarises these methods and the related applications.

2.3.1 Statistical and Empirical Analysis

Whereas statistics can broadly be defined as the discipline that deals with the collection; organisation; analysis; interpretation and presentation of data, empirical analysis refers to the research that uses empirical evidence (Romijn, 2014). Considering that using statistical methods in scientific studies is critical to determining the validity of empirical research, statistical methods and empirical studies have

TABLE 2.2: Summary of references related to the applications of scholarly data mining

Applications	Sub-Applications	References
Citation analysis	Citation and co-citation analysis	(Dey et al., 2017; Shi et al., 2015; Zehra and Umut, 2018; Ying et al., 2014) (Shu Zhao et al., 2018a; Jeong et al., 2014) (Trujillo and Long, 2018; Cornelia et al., 2015; Acuna et al., 2012) (An et al., 2017)
	Readership analysis	(Aduku et al., 2017; Maflahi and Thelwall, 2018; Thelwall, 2018)
	Bibliometrics analysis	(Monroy and Diaz, 2018; Godin, 2006; McBurney and Novak, 2002; Lv et al., 2011; Martínez-Gómez, 2015; Pilkington, 2004; Gleason, 1961; Merediz and Aurelio, 2019; Mingers and Leydesdorff, 2015)
	Altmetrics analysis	(Bornmann and Haunschild, 2018; Nabout et al., 2018; Priem and Costello, 2010; Weller et al., 2011)
Document analysis	Authorship analysis	(Rexha et al., 2018; Sun et al., 2011)
	Structure analysis	(Boyack et al., 2018; Heffernan and Teufel, 2018a; Lu et al., 2018a; Weber and Gunawardena, 2011)
	Content analysis	(Cornelia et al., 2015; Shardlow et al., 2018; Tuarob et al., 2016; S. Kim et al., 2018)
	Scientific recommendation	(Alam and Ismail, 2017; Shu Zhao et al., 2018a)
	Scientific text summarisation	(Mei and Zhai, 2008; Bhatia and Mitra, 2012; Rahul et al., 2013; Qazvinian and Radev, 2008)
Conference analysis		(Effendy et al., 2014; Effendy and Roland, 2016)
Trend analysis		(An et al., 2017; Hou et al., 2018; Rossetto et al., 2018; Soriano et al., 2018; C. Zhang and Guan, 2017) (Dridi et al., 2019a; Dridi et al., 2019b; Weismayer and Pezenka, 2017) (Effendy and Yap, 2017; Hoonlor et al., 2013)
Literature analysis		(Y. Liu et al., 2015; Tan et al., 2016; Tang, 2016; Tang et al., 2008; H. Li et al., 2006; Dunne et al., 2012; Osborne et al., 2013)

been widely used together in scientific research. Defined as "research about research", scholarly data mining has been particularly relying on statistical and empirical analysis mainly for citation analysis (Bornmann and Daniel, 2008; Bornmann and Haunschild, 2018; Virginia, 1989; Godin, 2006; McBurney and Novak, 2002; Lv et al., 2011; Martínez-Gómez, 2015; Monroy and Diaz, 2018; Nabout et al., 2018; Pilkington, 2004; Priem and Costello, 2010; Shadish et al., 1995; Thelwall, 2018; Weller et al., 2011; Acuna et al., 2012). This is justified by the quantitative aspect provided

by citation counts; they are measurable indicators of research impact. The quantitative aspect of citation counts has been used from different perspectives. The first perspective concerns the study of the scientific production. For instance, (Lv et al., 2011) have applied statistical analysis to evaluate global scientific production and developing trend of graphene research using the Science Citation Index, the Conference Proceeding Citation Index and the Derwent Innovation Index database integrated by Thomson Reuters databases. Similarly, (Martínez-Gómez, 2015) have applied statistical and predictive analyses to 286 scientific works published between 1973 and 2013 in order to study the evolution of the research and the dissemination of knowledge. In the same context, (Acuna et al., 2012) have relied on statistics to track scientific careers and predict scientific success using h-index. They have used a dataset of 3,085 neuroscientists, 57 Drosophila and 151 evolutionary scientists to understand how science develops. However, (Monroy and Diaz, 2018) have used statistics to study the dynamics of scientific production of several countries in terms of published papers. They have analysed Scopus database to identify dynamical changes that affected global scientific production such as social, political and economic circumstances. The second perspective concerns the study of the broad impact measurements of research beyond science, which is defined as *altmetrics* (Priem and Costello, 2010; Weller et al., 2011). In this context, (Nabout et al., 2018) have studied a dataset of 2,863 papers published in five ecological journals to study the correlation between traditional citation-based indicators and activities on online social media platforms such as Twitter and Mendeley. Similarly, (Bornmann and Haunschild, 2018) have studied Twitter data to measure the impact of science in order to fulfill the demands from governments and funding organisations. In addition to the statistical analysis, empirical and descriptive analyses have been used thoroughly (i) to study the origins of bibliometrics (Godin, 2006) and their purposes (McBurney and Novak, 2002); (ii) to study the citation behavior of scientists (Bornmann and Daniel, 2008; Virginia, 1989) and explore the meanings of citations (Shadish et al., 1995); and (iii) to investigate the intellectual pillars of the technology management literature and explore differences in the research agendas of worldwide scholars (Pilkington, 2004).

Some other scholarly data mining applications have been realised with statistical and empirical studies, such as trend analysis (Kaempf et al., 2015) and literature analysis (Wu et al., 2014). For trend analysis, the authors in (Kaempf et al., 2015) followed a statistical analysis to measure a topic importance based on page-view time series of Wikipedia articles. They have studied the emergence and life cycle of the emerging Hadoop market. To do so, they have developed *ETOSHA*, an open source software framework for Wikipedia analysis. ETOSHA has been used to investigate the changes in the frequency of views of Wikipedia pages. These changes have been used as indicator of collective interests and social trends. More specifically, the statistical analysis follows both qualitative interpretation and quantitative measurement of the network properties of Wikipedia pages. This includes measuring the context sensitive relevance of Wikipedia topics with respect to local and global neighborhood. As a matter of fact, ETOSHA has initially relied on exploratory data analysis (Tukey, 1977), namely representation plots, to unveil existing implicit semantic relationships between Wikipedia pages to automatically discover the context neighborhoods. Then, based on these neighborhoods, ETOSHA has used relative relevance indexes including the time-dependant relevance index that identifies content relevance and public recognition of Wikipedia topics. Unlike Google search that fails to reveal how other keywords with strong relation influence trends, ETOSHA

has leveraged context neighborhoods from Wikipedia page links to detect emerging trends. However, for literature analysis, descriptive statistics have been used to mine scholarly documents in a large-scale setting and provide scholarly applications, such as citation recommendation, expert recommendation and collaborator discovery. For instance, (Wu et al., 2014) have built a scholarly big data platform based on CiteseerX system that integrates different services for scholarly data such as information extraction and user/log data analytics. The proposed platform is based on a virtual architecture using a private cloud with the design of the key modules, which included a focused crawler, a crawl-extraction-ingestion workflow, and distributed repositories and databases.

2.3.2 Social Network Analysis

Due to the inherent social network generated from academic activities (such as citations, collaborations and academic communications) – named *academic social network* (Mumtazimah et al., 2018), social network analysis has been proposed to investigate the topologies and dynamics of this network (Feng et al., 2017). Social network analysis is mainly based on the graph theory (Deo, 1974) and aims to describe, analyse, and simulate an academic social network by representing, visualising and detecting communities in a given network of main scientific entities such as researchers, papers, conferences and citations. For instance, the *citation network* has been extensively studied to grasp the relationship among the scientific literatures (Rossetto et al., 2018). As well as, it has been used to detect the most influential nodes for graph summarisation problem on citation networks (Shi et al., 2015), and to study the power-law link strength distribution in paper co-citation networks (Star Zhao and Ye, 2013). The *paper network* has been used to aid in the exploration of relationships among scientific documents for different purposes. For instance, (Dunne et al., 2012) aimed to provide a summary, while identifying key papers, topics and research groups. To this end, they have developed Action Science Explorer (ASE) and have tested it on a collection of 17,610 computational linguistics papers from the ACL Anthology Network. On the other hand, (H. Li et al., 2006) have proposed CiteSeerX, which is a scientific literature library and search engine that automatically crawls and indexes scientific documents in the field of computer and information science. In addition to the paper network that studies scientific papers externally based on their interconnections via citation network, *topic network* tends to study the papers internally by studying the topics they are discussing from different perspectives. For instance, (Tao et al., 2017) have proposed a study map oriented method called Reference Injection based Double-Damping Page Rank (RIDP) that guides researchers to dig into the underlying principles of a specific paper. However, (S. Kim et al., 2018; Salatino et al., 2017) have studied the keywords associated with each paper in order to analyse the dynamics of research topics and visualise information on the growth and change in focus of research fields. The *academic network* has been studied by (Tang et al., 2008) to extract and mine academic social networks. They have provided search services for the academic network by extracting nearly half million research profiles.

Due to the strong relatedness between the aforementioned academic entities, social network analysis has been widely used to understand the large and heterogeneous networks formed by these entities and grasp the big picture of academic fields. Therefore, some works (Hoonlor et al., 2013; Osborne et al., 2013; Rossetto et al., 2018; Tan et al., 2016; Tang, 2016) have provided a systematic modeling approaches

to gain a deep understanding of the large academic networks. For instance, (Tang, 2016) have developed *AMiner* – based on a large scholar dataset with more than 130,000,000 researchers' profiles and 100,000,000 papers from multiple publication databases – in order to study the heterogeneous networks formed by authors, papers they have published, and venues in which they were published. In the same context, (Osborne et al., 2013) have developed *Rexplore*, which integrates statistical analysis, semantic technologies and visual analytics, to understand the dynamics of research areas, relate authors semantically, and perform fine-grained academic expert search along multiple dimensions.

It is worthy of note that the new learning paradigm *network representation learning* (D. Zhang et al., 2018) has recently attracted some works in big scholarly data due to its ability to capture complex relationships across various disciplines such as citation networks. In this respect, (Kong et al., 2018) have learned vector representation of papers with network embedding after bridging text information and structural identity with citation network, aiming to develop a robust scientific paper recommendation system. In other respects, (Jiaying Liu et al., 2019) have proposed a novel model that relies on network representation learning to discover advisor-advisee relationships hidden behind scientific collaboration networks.

Social network analysis has been widely used in scholarly data mining applications including citation analysis (Rossetto et al., 2018; Shi et al., 2015; Star Zhao and Ye, 2013), literature analysis (Tan et al., 2016; Tang et al., 2008; Tan et al., 2016; Tao et al., 2017; Osborne et al., 2013; Dunne et al., 2012; H. Li et al., 2006), document analysis (S. Kim et al., 2018; Salatino et al., 2017), conference analysis (Effendy et al., 2014) and trend analysis (Hoonlor et al., 2013; C. Zhang and Guan, 2017).

2.3.3 Machine Learning Techniques

Scholarly data mining involves different machine learning (ML) techniques ranging from supervised approaches to unsupervised approaches, namely classification and clustering.

Classification

In machine learning, classification refers to the task that requires the use of supervised learning algorithms to learn how to categorise a given set of data into classes (Alpaydin, 2010). Considering that the variety of scholarly data intrigues categorisation, classification has been used for different scholarly applications including (i) content-based citation analysis (Zehra and Umut, 2018), where citations were divided into four main categories; citation meaning, citation purpose, citation shape, and citation array; (ii) early identification of sleeping beauties – scientific publications, which do not get much cited for several years after being published, but then suddenly start getting cited heavily (Dey et al., 2017); (iii) paper reviewer recommendation (Shu Zhao et al., 2018a); (iv) identification of ambiguous author names (Sun et al., 2011); and (v) topic classification (Cornelia et al., 2015).

Different classification techniques have been used. For instance, (a) *Naîve Bayes Multinomial* and *Random Forest* algorithms have been used for automatic citation sentence classification in a dataset of 423 peer-reviewed articles associated with 12,881 references and 101,019 sentences, and have performed 90% success rate (Zehra and Umut, 2018). (b) *Linear Support Vector Machine*, *Decision Tree* and *KNN* have been

used to classify papers as sleeping beauties or not (Dey et al., 2017). The classifiers have been applied to a dataset of more than 2 million papers published in the computer science domain and indexed by Microsoft Academic Search; and have achieved a precision of 73% in identifying sleeping beauties immediately after their year of publications. In a different task, *Support Vector Machine* and *Naîve Bayes Multinomial* have been used for topic classification of research papers (Cornelia et al., 2015), applied to a subset sampled from the *CiteSeer^x* digital library. (c) *Logistic Regression* algorithm has been proposed to detect ambiguous author names in crowdsourced scholarly data extracted from Scholarmeter (Sun et al., 2011). Two classes of features of a scholar's publications are supplied to the classifier: (i) name variations and citations, and (ii) topic consistency, which helped to reach a 75% accuracy. In addition to the existing classification techniques, (Shu Zhao et al., 2018a) have proposed a novel classification method named (d) *Word Mover's Distance Constructive Covering Algorithm (WMD-CCA)* to solve the reviewer recommendation problem as a classification issue. It has been applied to four public datasets and a synthetic dataset from *Baidu Scholar*[31] and has shown its effectiveness to solve the reviewer recommendation task as a classification issue and improve the recommendation accuracy.

Clustering

Clustering, in machine learning, relies on unsupervised learning algorithms to divide data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups (Alpaydin, 2010). Due to the availability of unlabeled scholarly data, clustering has been used in different scholarly applications. As a matter of fact, clustering has been extensively used for document analysis (Ashton et al., 2012; Lu et al., 2018a; Salatino et al., 2018). For instance, the authors in (Ashton et al., 2012) and (Salatino et al., 2018) have relied on clustering to group research topics. The former study has grouped topics into clusters based on how authors move through them, while the later has detected clusters of topics that exhibit dynamics correlated with the emergence of new research topics. On the other hand, the authors in (Lu et al., 2018a) have used clustering for document structure analysis; they have generated domain-specific structures based on high-frequency section headers in scientific documents of a domain.

Similarly to its utility in document analysis, clustering has been used in literature analysis to explore internal structure of papers and finding research topics (Y. Liu et al., 2015). Furthermore, for trend analysis, clustering has been utilised to identify features of meta-knowledge (C. Zhang and Guan, 2017), and to investigate changes over time in the research landscape through clustering bursty keywords (Hoonlor et al., 2013).

Clustering has also served as a useful method for citation analysis (Dunne et al., 2012; Hou et al., 2018). This has been shown in the case of clusters identification of citation patters, which helps scholars by providing some forms of automated descriptions for interesting subsets of a document collection.

Different clustering techniques have been explored including *hierarchical clustering* (Ashton et al., 2012), *k-means* (Lu et al., 2018a) and *advanced clique percolation method (ACPM)*, which is a novel clustering algorithm developed by (Salatino et al.,

[31] https://scolary.com/tools/baidu-scholar

2018) to detect clusters of topics in the evolutionary networks that exhibit an intensive activity in terms of pace of collaboration.

Other ML Techniques

Other than classification and clustering, different other ML techniques have been used for scholarly data mining. For instance, *Hidden Markov Model* (HMM) – which is defined as a statistical tool that models generative sequences that can be characterised by an underlying process generating an observable sequence (Baum and Petrie, 1966) – has been explored for citation analysis. In (An et al., 2017), the authors identified dynamic patterns of knowledge flows driven by business method patents using HMM and patent citation data as an input. They have conducted a case study with the business method patents in 16 sub-classes related to secure transactions. Their analysis revealed that business method patents play increasingly important roles in advancement of business models. The proposed HMM based approach outperformed the existing research on knowledge flows that mainly focuses on static analysis while knowledge flows are intrinsically a dynamic phenomenon. Moreover, for document analysis, *ensemble learning* – which is a machine learning paradigm, where multiple learners are trained to solve the same problem (Polikar, 2006) – and *association rules* – which is a rule-based machine learning method, used to find correlations and co-occurrences between data sets (Piatetsky-Shapiro, 1991) – have been used by (Tuarob et al., 2016) to extract algorithm representations in a heterogeneous pool of scholarly documents. The proposed techniques discover pseudo-codes and algorithmic procedures, identify sections in scholarly documents, and use a heuristic that links different algorithm representations referring to the same algorithm together. The proposed techniques cover the limitations of the rule-based method proposed by (Bhatia et al., 2010) for pseudo-code detection, which assumes that each pseudo-code is accompanied by a caption. However, such an assumption is not usually true because of the wide variations in writing styles followed by different journals and authors. However *regression* – which is defined as a set of statistical processes that attempt to determine the strength and character of the relationship between one dependent variable and a series of other variables – has been used by (Asooja et al., 2016) to predict future keyword distribution in order to map scientific topic evolution over time. The prediction is based on historical data of $55k$ keywords extracted from Language Resources Evaluation Conference (LREC) conference proceedings from 2000 to 2014, and a time series dataset of topics and their popularity have been generated. Unlike existing approaches that simply map the evolution of scientific topics over years, the proposed approach automatically predicts keyword distribution. Consequently, it outperforms the methods based on topic modelling or clustering that require expert knowledge to manually label topics.

Deep learning – which is a form of machine learning based on artificial neural networks, which are capable to learn from unstructured and unlabelled data without human supervision – has been also used to analyse scientific literature. For instance (Safder and Hassan, 2018) have designed a deep search system for algorithms from full-text scholarly big data. In contrast to traditional term frequency-inverse document frequency (TF-IDF) based approaches that use frequent terms as in bag of words models, the authors first generated a synopsis of the full-text document and then enriched it with sentences that classify as algorithm-specific metadata from full-text to improve the capabilities of algorithmic-specific searching tasks. These

sentences were classified from deep learning based bi-directional long short term memory network (LSTM) model. The proposed model outperformed Support Vector Machine (SVM) in classifying 37,000 algorithm-specific metadata sentences with 81% accuracy.

2.3.4 Natural Language Processing Techniques

Scholarly data mining involves scholarly text mining (Feng et al., 2017), which plays an important role in the analysis of document content. Thus, text mining and natural language processing techniques have been widely employed to analyse scientific publications.

Current research in scholarly text mining relies mainly on topical analysis. Indeed, *topic model* – which is defined as a statistical model for discovering the abstract topics that occur in a collection of documents (Blei, 2012) –, namely *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003), has been extensively used either to assign topics to documents based on a given keyword set (document classification) (Paul and Girju, 2009; Tang et al., 2008; Weismayer and Pezenka, 2017) or to detect groups of similar documents (document clustering) (Ashton et al., 2012; Bakarov et al., 2018; Hall et al., 2008; Tang, 2016).

On the other hand, few recent works have explored *word embeddings* (Mikolov et al., 2013d) – the newly discovered natural language processing technique that represents individual words as real-valued vectors in a predefined vector space – to analyse the content of scientific publications. For instance, the authors in (J. He and Chen, 2018) have proposed word embeddings to track the semantic changes of scientific terms over time in the biomedical area. Going beyond the existing studies on topic-level analysis based on topic modeling techniques (Blei et al., 2003) – that automatically detect research topics based on textual information and identify their novelty –, the proposed approach investigates the impact of the novelty degree of research topics on the growth of scientific knowledge. In (Vahe et al., 2019), the authors have relied on word embeddings to capture latent knowledge from materials science literature and predict novel thermoelectric compositions. The authors have shown that – unlike supervised natural language processing techniques (Friedman et al., 2001; Swain and Cole, 2016), which requires large hand-labelled datasets for training – word embeddings can be efficiently used to encode materials science knowledge present in the published literature as information-dense vector representations without human labelling or supervision. As a result, without any explicit insertion of chemical knowledge, these embeddings capture complex materials science concepts such as the underlying structure of the periodic table and structure–property relationships in materials. Similarly, in this thesis (Dridi et al., 2019a; Dridi et al., 2019b) word embeddings have been used to early detect converging keywords that may result in trending topics in the area of machine learning.

2.4 Publications Areas

Scholarly data mining has been applied to a wide range of disciplines ranging from neuroscience (Acuna et al., 2012) to literature studies (Martínez-Gómez, 2015).

Figure 2.3 presents distribution per application domains of scholarly data mining applications. What can be clearly seen from this figure is that computer & information science is the field with greatest number of publications. This observation

TABLE 2.3: Summary of scholarly data mining methods and corresponding applications

		Citation analysis	Doc. analysis	Conf. analysis	Trend analysis	Literature analysis
Stat. & Emp. analysis		✓			✓	✓
SNA		✓	✓	✓	✓	✓
ML	Classification	✓	✓			
	Clustering	✓	✓		✓	
	HMM	✓				
	Ensemble Learning		✓			
	Association Rules		✓			
	Regression		✓			
	Deep Learning		✓			
NLP	Topic Models		✓		✓	
	Word Embeddings		✓		✓	

is somehow obvious because the majority of scholars investigating scholarly data mining are coming from the area of computer science, where the investigation of their area of expertise is more convenient for interpretation and conclusion drawing. Different sub-areas of computer science have been studied such as artificial intelligence (Alam and Ismail, 2017; Dridi et al., 2019a; Dridi et al., 2019b), computational linguistics (Ashton et al., 2012; Hall et al., 2008; Bakarov et al., 2018; Paul and Girju, 2009; Asooja et al., 2016), and big data (Kaempf et al., 2015). Besides, different scholarly data applications have been explored within the area of computer & information science such as document analysis (Alam and Ismail, 2017; Ashton et al., 2012; Salatino et al., 2017; Salatino et al., 2018; Bakarov et al., 2018; Paul and Girju, 2009; Tuarob et al., 2016; Cornelia et al., 2015), citation analysis (Zehra and Umut, 2018; Weller et al., 2011), literature analysis (Dunne et al., 2012; H. Li et al., 2006; Osborne et al., 2013), conference analysis (Effendy et al., 2014; Effendy and Roland, 2016; Tao et al., 2017; Nuzzolese et al., 2016), and trend analysis (Effendy and Yap, 2017; Dey et al., 2017; Asooja et al., 2016; Kaempf et al., 2015; Dridi et al., 2019a; Dridi et al., 2019b).

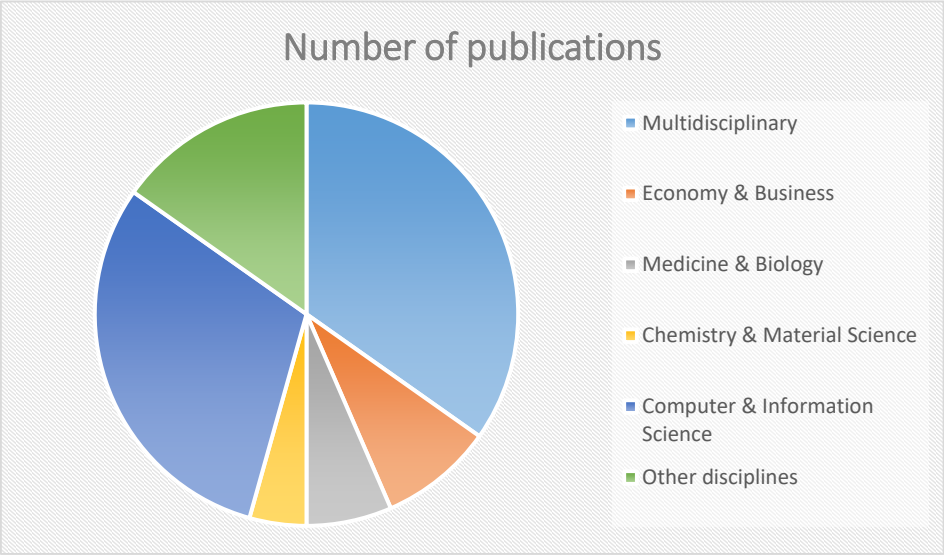


FIGURE 2.3: Distribution per application domains of scholarly data mining publications

Based on Figure 2.3, the second major part of studies has applied scholarly data mining to multidisciplinary area, where more than one discipline has been studied (Thelwall, 2018; Boyack et al., 2017; Wu et al., 2014; C. Zhang and Guan, 2017; Sun et al., 2011; Godin, 2006; McBurney and Novak, 2002; Monroy and Diaz, 2018; Tang et al., 2008).

Economy & Business area has also attracted the attention of scholars in scholarly data mining. They have investigated different aspects such as relations and economy (Soriano et al., 2018); innovation and entrepreneurial ecosystem (C. Zhang and

Guan, 2017); business (Rossetto et al., 2018) and business model innovation (An et al., 2017); and marketing and tourism (Weismayer and Pezenka, 2017).

The area of Medicine & Biology has been also studied through scholarly data mining. The existing studies (Rexha et al., 2018; Shardlow et al., 2018; Y. Liu et al., 2015) have studied the document content of biomedical publications to extract knowledge from scientific literature and discover underlying interesting research topics. Similarly, the area of Chemistry & Material Science has seen the application of scholarly data mining for citation analysis (Lv et al., 2011) and trend analysis (Vahe et al., 2019).

The other disciplines that have been involved in scholarly data analysis are as following: neuroscience (Acuna et al., 2012); ecology (Nabout et al., 2018), social science (Priem and Costello, 2010); education (Paul and Girju, 2009; Weber and Gunawardena, 2011); and translation and interpreting studies (Martínez-Gómez, 2015). The main scholarly data mining application applied to these areas is citation analysis.

2.5 Discussion

The majority of the reviewed studies demonstrated that scholarly data mining can be effectively applied to a wide range of scholarly applications to learn about the structure and the dynamics of science. The investigation done within this thesis suggested that scholarly data mining can be utilised to address different scholarly applications and provide better services to scholars such as academic recommendation, scientific text summarisation and research trend prediction. This is significant because these services can potentially accelerate science and facilitate the identification of fundamental mechanisms responsible for scientific discovery (Fortunato et al., 2018). However, despite its notable advantages, scholarly data mining also brings different challenges. The challenges that this thesis addresses are as follows.

Representation of scholarly data. Given the size of scholarly data, the complexity of its structure and the nature of the scientific language, the representation of the scientific content has become increasingly challenging. In fact, big scholarly data is characterised by the 5V feature (volume, variety, velocity, value, veracity) (Feng et al., 2017). Velocity refers to the dynamics of scholarly data, including the scientific language. This velocity reflects therefore the dynamics of science. This feature makes from the representation of scholarly content a challenging task. This thesis addresses this challenge by focusing on the representation of the scientific language as an important type of scholarly data. To this end, temporal word embeddings are leveraged to represent the scientific text towards a better tracking of the dynamics of science. A rigorous study has been conducted to test the effectiveness of word embeddings to represent the semantics behind the scientific language. This study is described in Chapter 4.

Lack of methodologies and concepts. The computational history of science, as an application of scholarly data mining, is increasingly attracting research interest. However, no rigorous methodologies are available as standards to follow to perform the computational history of science. The existing works track

the evolvement of research topics or perform the trend analysis, claiming performing the computational history of science. While trend analysis is a key step towards the computational history of science, the concept of *trend* is still ambiguous; no standard definition is given. Furthermore, the computational history of science is not only limited to the task of trend analysis. It rather includes the tracking of the dynamics of science. Establishing methodologies and defining concepts to perform an accurate computational history of science is therefore a need. This thesis attempts to satisfy this need by proposing a methodology that includes the detection of emerging trends (Chapter 5 and 6) and the tracking of the dynamics of science (Chapter 7). Importantly, in this thesis, the concept of *trend* is defined, and accordingly the trends are detected (Chapter 5 and 6).

Lack of gold standards. Some scholarly applications require gold standards to evaluate their outcomes such as trend analysis. However, there are no standards to use to perform comparative studies or to validate the obtained results. Most of existing studies on trend analysis have relied on descriptive analysis to present their studies. However, applying machine learning techniques requires standards for validation, which makes this task challenging. This thesis attempts to build a gold standard relying on Google Trends hits. The approach presented in Chapter 6 highlights the importance of promoting gold standards for the matter of trend analysis. In fact, this scholarly application represents an important direction towards knowledge discovery and the study of the dynamics of science.

Scholarly data mining brings some other challenges that are not addressed in this thesis. These challenges are described as follows.

Collecting and processing scholarly data. Given the 5V feature (volume, variety, velocity, value, veracity) (Feng et al., 2017), veracity and variety make from collecting and processing scholarly data a complex task, where ambiguity is present and different entities are involved. The complexity of this task makes scholarly data management a challenging task.

Insufficiency of metrics to evaluate the research quality. Evaluating the research quality is an essential component of research assessment, and outcomes of such evaluations can help in institutional research strategies such as funding and recruitment. However, there are little standards to measure scientific performance objectively. For instance, metrics alone have been unable to achieve the task of predicting scientific impact and assessing research quality (Sahel, 2011). Improving existing research evaluation practices is, therefore, an urge.

2.6 Summary

This chapter has provided a systematic review about scholarly data mining applications, performing a literature-based analysis, and a description of current approaches investigating scholarly data; indicating the interest in the field from different perspectives such as the type of the techniques used and the disciplines investigated.

The value proposition of scholarly data mining is that with a deeper understanding of the structure of science, more scientific discovery problems can be effectively addressed, and tools and policies that have the potential to perform computational history of science can be developed. In the following chapters, the proposed solution to tackle the problem of the computational history of science is presented. This solution relies mainly on *word embedding* techniques, namely *word2vec* (Mikolov et al., 2013c). For this reason, the next chapter will introduce the foundations of word embedding techniques and describe the architectures of word2vec.

Chapter 3

Word Embeddings Techniques – Word2vec

> "The knowledge of many minds consists principally of the news of the day and the talk at the last tea-party."
>
> — James Lendall Basford, *it Sparks from the Philosopher's Stone*. (1845–1915)

As discussed in the previous chapter, a natural language processing (NLP) technique, namely *word embeddings*, has been used for the analysis of scholarly data. This chapter details *word embeddings* that represent the NLP technique used throughout this thesis. In this chapter, after stating the history of word embeddings in Section 3.1, the three main foundations of word embeddings are represented in Section 3.2, namely *vector space semantics*, *word senses*, and *machine learning*. Then, *word2vec* (Mikolov et al., 2013d) – the word embedding technique used in this thesis – is extensively described in Section 3.3 by describing its two models in Section 3.3.1. Afterwards, a summary of works on *temporal word embeddings* is given in Section 3.4. This is justified by the fact that the approaches proposed in this thesis will adopt *temporal word embeddings* to perform a computational history of science. Finally, other word embedding techniques are described in Section 3.5, where also a discussion section (Section 3.5.4) is provided to justify the choice of word2vec in this thesis as a word embedding technique to perform the computational history of science. Finally, Section 3.6 summarises the chapter.

3.1 History of Word Embeddings

In Computational Linguistics, word embeddings (WEs) have a long history in the area of Distributional Semantics, where the term *distributional semantic model* was dominating till 2013 when a team at Google led by Thomas Mikolov created *word2vec* (Mikolov et al., 2013e), a word embedding toolkit. And, since then, the terminology *word embedding (WE)* started to be the dominating term.

The basic idea of word embeddings is that the contextual information constitutes a viable representation of linguistic items. This idea has its theoretical root in language philosophy *"a word is characterized by the company it keeps"* (Firth, 1957). The earliest attempt at representing words as vectors dates back to the 1960s with the development of the *vector space model* (G. Salton et al., 1975) and *Osgood's semantic differentials* (Osgood et al., 1975) that use handcraft features. Methods that use automatically generated contextual features were developed around the 1990s and can

be divided into two categories: count-based methods (e.g., *Latent Semantic Analysis* (Furnas et al., 1988)), and predictive methods (e.g., *neural probabilistic language models* (Yoshua Bengio et al., 2003)).

The difference between these two categories of methods is mainly the type of contextual information they use. The count-based models use documents as contexts, which is justified by their roots in information retrieval. The predictive models instead use words as contexts taking into account the linguistic and the cognitive perspective.

The area developed gradually and has seen an explosion in 2013 when word2vec (Mikolov et al., 2013e) was developed. Since then, most new word embedding techniques rely on neural network architecture instead of n-gram models, which are statistical language models that assign probabilities to the sequences of words (Jurafsky and Martin, 2009).

3.2 Foundations of Word Embeddings

This section introduces the three main foundations of word embeddings, which are *vector space semantics*, *word senses* and *machine learning*.

3.2.1 Vector Space Semantics

Vector Space Semantics, also known as *Distributional Semantics* (DS) (Harris, 1954), is defined as a usage-based model of meaning (Lenci, 2018) that aims to distributionally represent words by means of vectors. The meaning of a word is supposed to be entirely defined by *"the company it keeps"* (Firth, 1957). Indeed, the vector space semantics rely on the *distributional hypothesis* (Harris, 1954) that underlines the idea that *"words that occur in similar contexts tend to have similar meanings"* (Erk, 2012; Turney and Pantel, 2010).

Distributional Semantics are typically implemented through *vector space models (VSM)*, where words are represented as points in high-dimensional space. The VSM have been introduced in the area of *information retrieval* by Salton *et al.* (G. Salton et al., 1975) and represent a collection of documents with a matrix whose rows are vectors corresponding to lexical items and whose columns are vectors corresponding to documents, and each matrix entry records the occurrences of a lexical item in a document (Lenci, 2018).

VSM gained a lot of success and continued to be used in information retrieval. However, they were completely ignored in computational linguistics until the early 1990s, because of the dominance of formal and logic approaches (Lenci, 2018). Starting in the late 1980s, NLP has seen a revolution with the introduction of machine learning (ML) algorithms for language processing. This revolution was termed *" statistical revolution"* (Johnson, 2009) as NLP research has relied heavily on ML. This favored a growing interest in DS that became a mainstream research paradigm in computational linguistics (Lenci, 2018).

DS represent meaning through observed contexts. Figure 3.1 shows a simple example for the target word *"brexit"*. The corpus on the left (*wiki corpus*) consists of few sentences from Brexit Wikipedia page. In the middle are co-occurrence counts derived from the corpus (showing counts for only some of the context words). For this example, the lemmatised context words are counted in the full sentence in which the target *" brexit"* occurs. DS are commonly implemented in vector space models

(VSM) that represent a target word – here, *"brexit"* – as a point in high-dimensional space. The dimensions correspond to context items, and in the simplest case, the coordinates are the co-occurrence counts. Figure 3.1 (right) shows parts of the vector of *"brexit"* derived from *wiki corpus*.

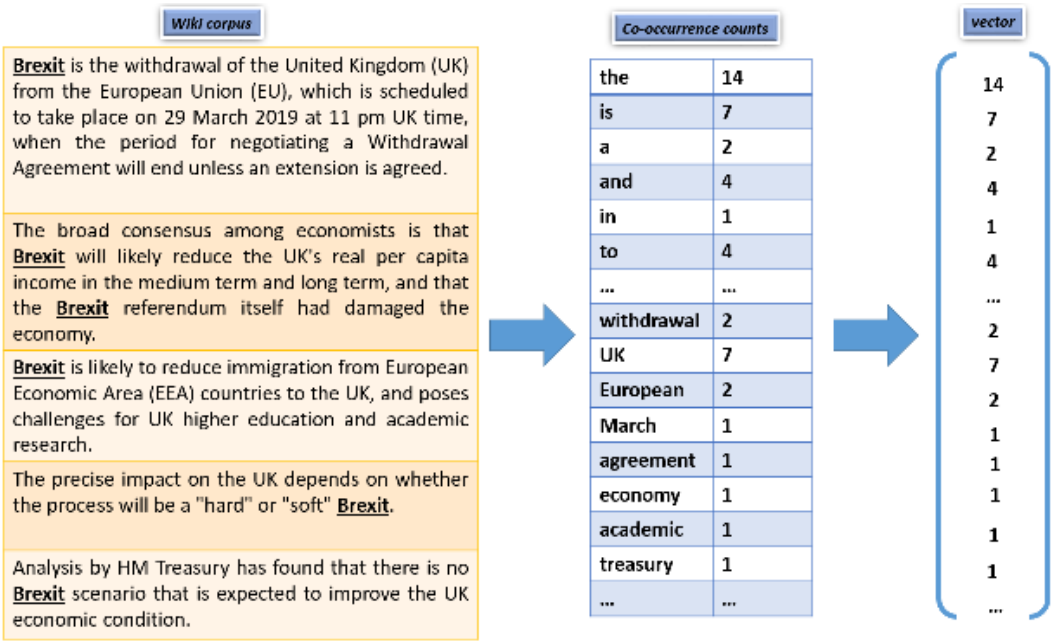


FIGURE 3.1: Creating a simple vector space representation for *"brexit"*: A wiki corpus of sample sentences from the Brexit Wikipedia page (left), context word counts (middle), and the corresponding vector (right)

Figure 3.1 explains the basic idea of the VSM that represent text as *Bag of Words (BoW)* using word frequencies. More formally, a vocabulary V is established, where each word w_i has a unique integer index i. A document d is represented by a column vector v_j, where each element v_{ij} stores the frequency f_{ij} of word w_i in document d_j.

For the standard vector space model, BoW is normalised such that each v_{ij} value does not necessarily show the exact word/term frequency, but stores a weight w_{ij} that represents a relevance measure of the term in the document. Some of the most popular weighting schemes are *TF.IDF* (Karen, 1988) and *BM25* (Robertson et al., 1992). Table 3.1 presents the known weighting schemes.

The distributional similarity between two words is measured with the similarity between their distributional vectors. The *cosine* is the most popular measure of vector similarity in DS. Table 3.2 summarises all similarity metrics between two vectors.

The Distributional Semantic Model (DSM) is a particular configuration of the parameters used to build distributional representations. These parameters include the selection of target lexemes, the definition of context type, the choice of weighting scheme, the application of dimensionality reduction, and the choice of a vector similarity measure (Lenci, 2018). One of the major variations among distributional semantic models is the method to learn distributional representations. Matrix models are the most common method of distributional semantic models. They generalise the vector space models and learn the representation of a target lexeme by recording its co-occurrences in linguistic contexts. Table 3.3 presents the most common matrix distributional semantic models.

TABLE 3.1: Term weighting schemes. f_{ij} denotes the target word frequency in a particular context, f_i the total target word frequency, f_j the total context frequency, N the total of all frequencies, n_j the number of non-zero contexts. $P(t_{ij}|c_j)$ is defined as $\frac{f_{ij}}{f_j}$ and $P(t_{ij})$ is defined as $\frac{f_{ij}}{N}$.

Weighting Schema	Definition
None (Harris, 1954)	$w_{ij} = f_{ij}$
TF.IDF (Karen, 1988)	$w_{ij} = \log(f_{ij}) \times \log(\frac{N}{n_j})$
TF.ICF (Reed et al., 2006)	$w_{ij} = \log(f_{ij}) \times \log(\frac{N}{f_j})$
Okapi BM25 (Robertson et al., 1992)	$w_{ij} = \frac{f_{ij}}{0.5+1.5\times\frac{f_j}{\frac{f_j}{j}}+f_{ij}} \log\frac{N-n_j+0.5}{f_{ij}+0.5}$
ATC (Sebastiani, 2002)	$w_{ij} = \frac{(0.5+0.5\times\frac{f_{ij}}{max_f})\log(\frac{N}{n_j})}{\sqrt{\sum_{i=1}^{N}[(0.5+0.5\times\frac{f_{ij}}{max_f})\log(\frac{N}{n_j})]^2}}$
LTU (Singhal et al., 1996)	$w_{ij} = \frac{(\log(f_{ij})+1.0)\log(\frac{N}{n_j})}{0.8+0.2\times f_i\times\frac{j}{f_i}}$
MI (Cover and Thomas, 2006)	$w_{ij} = \log\frac{P(t_{ij}\|c_j)}{P(t_{ij})P(c_j)}$
Lin98a (Lin, 1998b)	$w_{ij} = \frac{f_{ij}\times f}{f_i\times f_j}$
Lin98b (Lin, 1998a)	$w_{ij} = -1 \times \log\frac{n_j}{N}$
Gref94 (Grefenstette, 1994)	$w_{ij} = \frac{\log f_{ij}+1}{\log n_j+1}$

TABLE 3.2: Similarity measures between vectors v and u, where v_i is the i^{th} component of v

Measure	Definition
Cosine	$\frac{u \cdot v}{\lvert u \rvert \cdot \lvert v \rvert}$
Euclidean	$\frac{1}{1+\sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}}$
Cityblock	$\frac{1}{1+\sum_{i=1}^{n}\lvert u_i - v_i \rvert}$
Chebyshev	$\frac{1}{1+max_i\lvert u_i - v_i \rvert}$
Correlation	$\frac{(u-\bar{u})\cdot(v-\bar{v})}{\lvert u \rvert \cdot \lvert v \rvert}$
Dice	$\frac{2\sum_{i=0}^{n} min(u_i, v_i)}{\sum_{i=0}^{n} u_i + v_i}$
Jaccard	$\frac{u \cdot v}{\sum_{i=0}^{n} u_i + v_i}$
Jaccard2	$\frac{\sum_{i=0}^{n} min(u_i, v_i)}{\sum_{i=0}^{n} max(u_i, v_i)}$
Lin	$\frac{\sum_{i=0}^{n} u_i, v_i}{\lvert u \rvert + \lvert v \rvert}$
Tanimoto	$\frac{u \cdot v}{\lvert u \rvert + \lvert v \rvert - u \cdot v}$
Jensen-Shannon Div	$1 - \frac{\frac{1}{2}(D(u\lvert\lvert\frac{u+v}{2}) + D(v\lvert\lvert\frac{u+v}{2}))}{\sqrt{2\log 2}}$
α-skew	$1 - \frac{D(u\lvert\lvert \alpha v + (1-\alpha)u)}{\sqrt{2\log 2}}$

TABLE 3.3: The most common matrix distributional semantic models (Lenci, 2018)

Model	Description	Reference
Latent Semantic Analysis (LSA)	Word-by-region matrix, weighted with entropy and reduced with Singular Value Decomposition (SVD)	(Landauer and Dutnais, 1997)
Hyperspace Analogue of Language (HAL)	Window-based model with directed collocates	(Burgess, 1998)
Latent Relational Analysis (LRA)	Pair-by-pattern matrix reduced with SVD to measure relational similarity	(Landauer and Dutnais, 1997)
Dependency Vectors (DV)	Syntactic model with dependency-filtered collocates	(Pado and Lapata, 2007)
Topic Models	Wordr-by-region matrix reduced with Bayesian inference	(Steyvers and Griffiths, 2007)
Distributional Memory (DM)	Target-link-context tuples formalised with a high order tensor	(Baroni and Lenci, 2010)
High-Dimensional Explorer (HiDEx)	Generalisation of HAL with a larger range of parameter settings	(Shaoul and Westbury, 2010)
Global Vectors (GloVe)	Word-by-word matrix reduced with weighted least-squares regression	(Pennington et al., 2014a)

3.2.2 Word Senses

In linguistics, *word sense* is defined as the word usage as per Wittgenstein's suggestion *"the meaning of a word is its use in the language"* (Wittgenstein, 1953). The definition of *meaning* was more specified by Harris, when he proposed that words with similar syntactic usage have similar meaning (Harris, 1954).

Since words might have different meanings (word senses), people and computers must use a process called *word-sense disambiguation (WSD)* (Ide and Véronis, 1998) to find the correct meaning of a word. This process uses context (such as neighboring words) to narrow the possible senses down to the possible ones. Research on WSD has been an interest since the earliest days of computer treatment of language in the 1950's, where language understanding was required for different applications such as machine translation (Yehoshua, 1960), information retrieval (Gerard Salton, 1968; Gerard Salton and McGill, 1986), content and thematic analysis (P. J. Stone and Hunt, 1963; P. Stone, 1969), grammatical analysis (Jensen and Binot, 1987), speech processing (Sproat et al., 1992), text processing (Yarowsky, 1994).

There are four different approaches to WSD:

- **Knowledge based approaches.** They rely on several types of lexical knowledge bases such as dictionary or thesaurus, WordNet (Miller, 1995), SemCor (Landes et al., 1998), Wikipedia, *etc*., to provide the appropriate sense of word in a context. The fundamental principle of these approaches is the matching of information, obtained from the context of the word, with the information obtained from the lexical knowledge base (Singh and Gupta, 2015).

- **Supervised approaches.** They rely on the assumption that the context provides evidence to disambiguate words. In other words, the basic idea is that words surrounding the target word can provide clues about the word sense, these words are called *features*. These features are learned by a classifier on training data, and then the classifier is used on test data to see how much accurately the selected features have disambiguated the word sense (Singh and Gupta, 2015). The most commonly used algorithms in supervised approaches are Naive Bayes (Jin et al., 2010), Decision Trees (Quinlan, 1990), Neural Networks (McCulloch and Pitts, 1988) and Support Vector Machines (Boser et al., 1992).

- **Unsupervised approaches.** They overcome the problem of knowledge acquisition bottleneck because they do not require sense annotated data. They rely on the assumption that similar senses occur in similar contexts, and thus the important task of these approaches is to identify sense clusters (Chandra and Dwivedi, 2014) using similarity measures of context. Then, new occurrences of the word can be classified into the closest clusters or senses. Different unsupervised WSD approaches have been proposed such as HyperLex approach (Véronis, 2004), Roget's Categories approach (Yarowsky, 1992) and Lin's approach (Lin, 1997).

- **Semi-supervised approaches.** They are also known as *weakly supervised approaches*, which allows both labeled and unlabeled data. They tend to solve the problem when fully labeled data is not available and it is expensive and time consuming to label the unlabeled data. Some semi-supervised approaches have been proposed such as the *Yarowsky algorithm* (Yarowsky, 1995) and the *bootstrapping* approach (Yarowsky, 1994). The *Yarowsky algorithm* (Yarowsky,

1995) was an early example of such approaches. it uses the *"one sense per collocation"* and *"one sense per discourse"* properties of human languages for WSD. The *bootstrapping* approach (Yarowsky, 1994) starts with small amount of sense labeled data (seed data), a large amount of unlabeled data and one or more classifiers. The seed data is used to train an initial classifier, using any supervised algorithm. This classifier is then used on the unlabeled data to extract a larger training set, in which only the most confident classifications are included. The process is repeated, each new classifier being trained on a successively larger training set until the whole data set is trained.

3.2.3 Machine Learning

Machine learning is defined as a subset of artificial intelligence dealing with algorithms that allow computer programs to automatically improve through experience (Mitchell, 1997). Machine learning algorithms are then used to make predictions from training data without being explicitly programmed to do so (Koza et al., 1996). This advantage makes machine learning algorithms evolve and has brought about an explosion of use in different applications such as image recognition, NLP, automatic speech recognition, *etc*.

A subset of machine learning algorithms is related to *computational statistics*, while another subset is related to *neural networks* and their derivations. The first subset focuses on making predictions using computers, while the second tends to imitate the human brain in processing data and creating patterns for use in decision making. Both subsets of machine learning algorithms have been leveraged in NLP.

In this thesis, the focus is on the application of machine learning algorithms in NLP that has been started since the early 2000s with the implementation of language models (Yoshua Bengio et al., 2003), and recently with the introduction of *word embeddings* (Mikolov et al., 2013d).

3.3 Word Embeddings – Word2vec

This section describes the two main models of word2vec, which are *the continuous-bag-of-words model (CBOW)* and *the skip-gram model (SG)*.

3.3.1 Models of Word2vec

Introduced by Mikolov et *al.* (Mikolov et al., 2013e), word2vec was the first popular embeddings technique for NLP tasks. The model is used for learning vector representations for words, where semantically similar words are mapped to nearby points in the vector space. This vector representation is performed through the two-layer neural network that characterises word2vec. It takes a text corpus as input and produces word vectors as output. The embeddings are actually the weights of the hidden layer in the neural network.

Word2vec is defined as a computationally-efficient predictive model that either uses context to predict a target word (a model known as *CBOW* model), or uses a word to predict a target context (which is called *skip-gram* model) (Mikolov et al., 2013e). Figure 3.2 represents the difference between the two models.

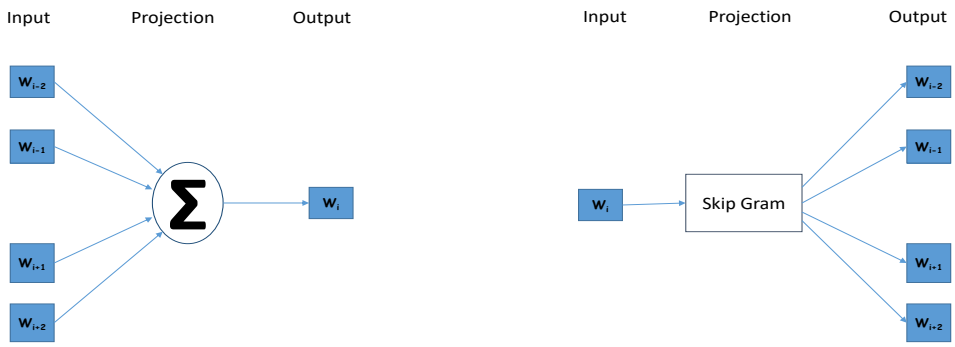


FIGURE 3.2: Word2vec architectures: CBOW and skip-gram models

Continuous Bag-of-Words Model

The CBOW model predicts target words from source context words (Mikolov et al., 2013e). The architecture of the CBOW model, shown in Figure 3.3, is detailed as follows: the input layer consists of the one-hot encoded input context words $\{x_1, \ldots, x_C\}$ for a word window of size C and vocabulary of size V. The hidden layer is an N-dimensional vector h. Finally, the output layer is the output word y in the training example, which also consists of a one-hot encoded vector. The one-hot encoded input vectors are connected to the hidden layer via a $V \times N$ weight matrix W and the hidden layer is connected to the output via a $N \times V$ weight matrix W' (*Word2Vec Tutorial Part II: The Continuous Bag-of-Words Model* n.d.).

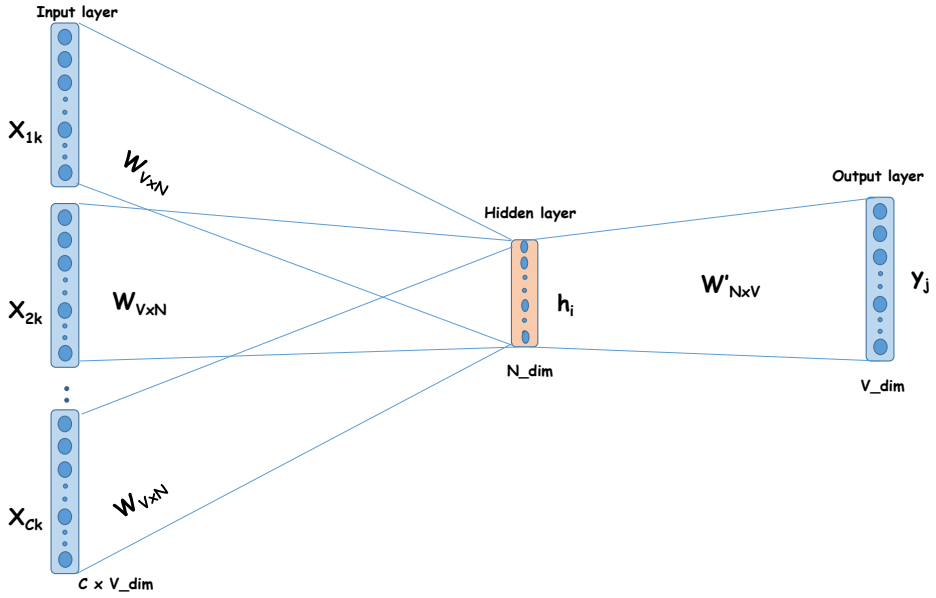


FIGURE 3.3: Continuous bag-of-words architecture

The output vectors are computed from the inputs via froward propagation. The first step consists to evaluate the output of the hidden layer h by computing the

average of the input vectors weight by the matrix W as follows:

$$h = \frac{1}{C} W \cdot (\sum_{i=1}^{C} x_i) \tag{3.1}$$

The second step consists to compute the inputs to each node in the output layer as follows:

$$u_j = v'^{T}_{w_j} \cdot h \tag{3.2}$$

where v'_{w_j} is the j^{th} column of the output matrix W'. At the final step, the output y_j of the output layer is computed by passing the input u_j throught the *softmax function* as follows:

$$y_j = P(w_{y_j}|w_1, \ldots, w_c) = \frac{exp(u_j)}{\sum_{j=1}^{V} exp(u'_j)} \tag{3.3}$$

In order to learn the weight matrices, W and W', randomly initialized values have to be chosen. Then, training examples have to be sequentially fed to the model while observing the error that represents the difference between the expected output and the observed output. The gradient of this error is computed with respect to the elements of both weight matrices W and W', and the errors are corrected in the direction of this gradient. The general optimisation procedure is known as *stochastic gradient descent (SGD)*, but the method by which the gradients are derived is called *backpropagation*.

The first step is to define the loss function. The objective is to maximize the conditional probability of the output word w_{output} given the input context w_{input}, therefore the loss function is defined in Equation 3.4.

$$E = -\log p(w_{output}|w_{input}) = -u_{j^*} - \log \sum_{j'=1}^{V} exp(u_{j'}) = -v^{T}_{w_{output}} \cdot h - \log \sum_{j'=1}^{V} exp(v^{T}_{w_{j'}} \cdot h) \tag{3.4}$$

where j^* is the index of the actual output word.

The next step consists to derive the update equation for the hidden-output layer weights W', then derive the weights for the input-hidden layer weights W.

To update the hidden-output layer weights, three steps have to be followed. The first step consists to compute the derivative of the loss function E with respect to the input to the j^{th} node in the output layer u_j.

$$\frac{\partial E}{\partial u_j} = y_j - t_j \tag{3.5}$$

where $t_j = 1$ if $j = j^*$ otherwise $t_j = 0$. This is simply the prediction error of node j in the output layer. The second step consists to take the derivative of E with respect to the output weight w'_{ij} using the chain rule.

$$\frac{\partial E}{\partial w'_{ij}} = \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial w'_{ij}} = (y_i - t_j) \cdot h_i \tag{3.6}$$

The third step consists to define the stochastic gradient descent equation, given the obtained gradient with respect to an arbitrary output weight w'_{ij}, as follows:

$$w'^{(new)}_{ij} = w'^{(old)}_{ij} - \eta \cdot (y_j - t_j) \cdot h_i \tag{3.7}$$

or

$$v'^{(new)}_{w_j} = v'^{(old)}_{w_j} - \eta \cdot (y_j - t_j) \cdot h \tag{3.8}$$

where $\eta > 0$ is the learning rate.

To update the input-hidden layer weights, similarly to the hidden-output layer, a similar update equation for the input weights w_{ij} has to be derived. The first step consists to compute the derivative of E with respect to an arbitrary hidden node h_i using the chain rule as follows:

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} (y_j - t_j) \cdot w'_{ij} \tag{3.9}$$

where the sum is needed because the hidden layer node h_i is connected to each node of the output layer and therefore each prediction error must be incorporated. The second step consists to compute the derivative of E with respect to an arbitrary input weight w_{ki} as follows:

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial h_i} \cdot \frac{\partial E}{\partial w_{ki}} = \sum_{j=1}^{V} (y_j - t_j) \cdot w'_{ij} \cdot \frac{1}{C} \cdot x_k = \frac{1}{C}(X \cdot EH) \tag{3.10}$$

where EH is an N-dimensional vector of elements $\sum_{j=1}^{V} (y_j - t_j) \cdot w'_{ij}$ from $i = 1, \ldots, N$. However, since the inputs X are one-hot encoded, only one row of the $N \times V$ matrix $\frac{1}{C}(X \cdot EH)$ will be nonzero. Thus, the final stochastic gradient descent equation for the input weights is given as follows:

$$v'^{(new)}_{w_{Input,c}} = v'^{(old)}_{w_{Input,c}} - \eta \cdot \frac{1}{C} \cdot EH \tag{3.11}$$

where $w_{Input,c}$ is the c^{th} word in the input context.

Skip-gram Model

Differently to CBOW model that predicts the current word based on the context, the *skip-gram* model uses each current word as an input to predict words within a certain range before and after the current word (Mikolov et al., 2013e). More formally, the input of the skip-gram model is a single word w_{Input} and the output is the words in the w_{Input}'s context $\{w_{Output,1}, \ldots, w_{Output,C}\}$ defined by a word window of size C.

The architecture of the skip-gram model is shown in Figure 3.4. X represents the one-hot encoded vector corresponding to the input word in the training instance and $\{y_1, \ldots, y_C\}$ are the one-hot encoded vectors corresponding to the output words in the training instance. The $V \times N$ matrix W is the weight matrix between the input layer and the hidden layer whose i^{th} row represents the weights corresponding to the i^{th} word in the vocabulary. The learning focuses on the weight matrix W because it contains the vector encodings of all of the words in the vocabulary (*Word2Vec Tutorial Part I: The SkipGram Model* n.d.).

Each output word vector also has an associated $N \times V$ output matrix W'. The hidden layer consists of N nodes, where the input to a unit in this layer h_i is the weighted sum of its inputs. Since the input vector X is one hot encoded, the weights coming from the nonzero element will be the only ones contributing to the hidden layer. Therefore, for the input X with $x_k = 1$ and $x_{k'} = 0$ for all $k' \neq k$ the outputs of the hidden layer will be equivalent to the k^{th} row of W. Or mathematically,

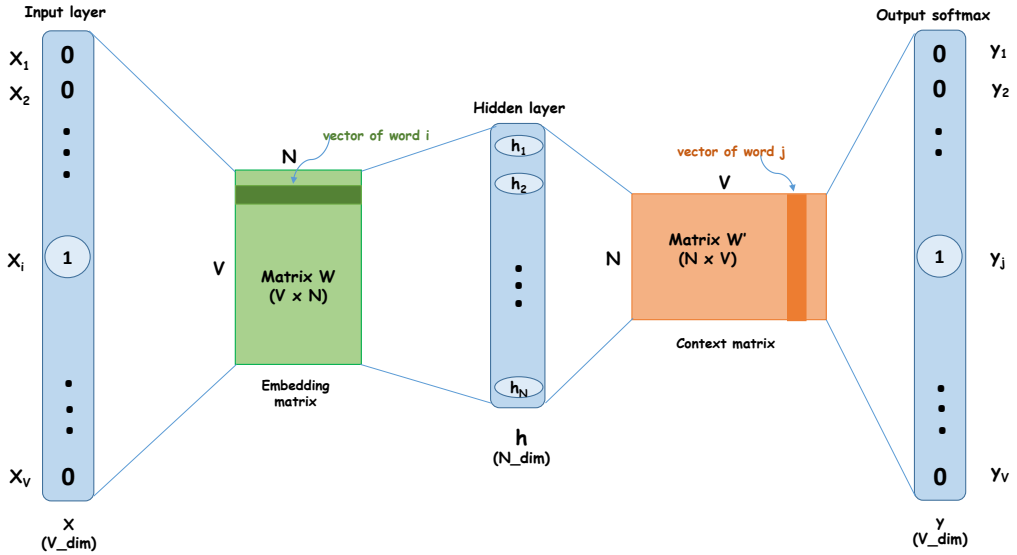


FIGURE 3.4: Skip-gram architecture

$$h = X^T W = W_{(k,.)} := V_{w_{Input}} \tag{3.12}$$

where no activation function is used here because the inputs are bounded by the one-hot encoding.

In the same way, the inputs to each of the $C \times V$ output nodes is computed by the weighted sum of its inputs. Therefore, the input to the j^{th} node of the c^{th} output word is

$$u_{c,j} = V'^T_{w_j} \cdot h \tag{3.13}$$

It is worth noting that the output layers for each output word share the same weights therefore $u_{c,j} = u_j$. Finally, the output of the j^{th} node of the c^{th} output word is computed via the softmax function, which produces a multinomial distribution.

$$p(w_{c,j} = w_{(Output,c)|w_{Input}}) = y_{c,j} = \frac{exp(u_{c,j})}{\sum_{j'=1}^{V} exp(u_{j'})} \tag{3.14}$$

where the obtained value represents the probability that the output of the j^{th} node of the c^{th} output word is equal to the actual value of the j^{th} index of the c^{th} output vector.

After these steps, the inputs are propagated forward through the network to produce outputs. The aim after is to derive the error gradients necessary for the backpropagation algorithm to learn both W and W'. To learn W', three steps are provided. The first step is defining a loss function as follows:

$$\begin{aligned} E &= -\log p(w_{Output,1}, w_{Output,2}, \ldots, w_{Output,C} | w_{Input}) \\ &= -\log \Pi_{c=1}^{C} \frac{exp(u_{c,j*_c})}{\sum_{j'=1}^{V} exp(u'_j)} \\ &= -\sum_{c=1}^{C} u_{j*_c} + C \cdot \log \sum_{j'=1}^{V} exp(u'_j) \end{aligned} \tag{3.15}$$

where the obtained value represents the probability of the output words (the words in the input word's context) given the input word w_{Input}, and $j*_c$ is the index of the c^{th} output word. The second step consists to compute the error derivative with respect to the inputs of the final layer $u_{c,j}$ as follows:

$$\frac{\partial E}{\partial u_{c,j}} = y_{c,j} - t_{c,j} \tag{3.16}$$

where $t_{c,j} = 1$ if the j^{th} true output word is equal to 1, otherwise $t_{c,j} = 0$. This represents the prediction error of the j^{th} node of the c^{th} output word.

Once the error derivative with respect to $u_{c,j}$ is found, the third step consists to derive the derivative with respect to the output matrix W' using the chain rule as follows:

$$\frac{\partial E}{\partial W'_{ij}} = \sum_{c=1}^{C} \frac{\partial E}{\partial u_{c,j}} \cdot \frac{\partial u_{c,j}}{\partial w'_{ij}} = \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) \cdot h_i \tag{3.17}$$

Therefore, the gradient descent update equation for the output matrix W' is given by:

$$w'^{(new)}_{ij} = w'^{(old)}_{ij} - \eta \cdot \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) \cdot h_i \tag{3.18}$$

Similarly to W', three steps are required to derive the update equation for the input-hidden layer weights in W. The first step consists to compute the error derivative with respect to the hidden layer as follows:

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) \cdot w'_{ij} \tag{3.19}$$

The next step computes the derivative with respect to W following Equation 3.20.

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}} = \sum_{j=1}^{V} \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) \cdot w'_{ij} \cdot x_k \tag{3.20}$$

Finally, the gradient descent equation for the input weights is given as following:

$$w^{(new)}_{ij} = w^{(old)}_{ij} - \eta \cdot \sum_{j=1}^{V} \sum_{c=1}^{C} (y_{c,j} - t_{c,j}) \cdot w'_{ij} \cdot x_j \tag{3.21}$$

Each gradient descent update requires a sum over the entire vocabulary V, which is computationally expensive. In order to make this computation more efficient,

computation techniques such as *hierarchical softmax* and *negative sampling* are used in practice.

Hierarchical Softmax

It was introduced by Morin and Bengio (Morin and Y. Bengio, 2005) in the context of neural network language models and provides a computationally efficient approximation of the full softmax. In order to obtain the probability distribution of V output nodes in the neural network, instead of evaluating all nodes it evaluates only about $\log_2(V)$ output nodes (Mikolov et al., 2013d).

The hierarchical softmax defines a binary tree representation of the output layer with the V words as leaves and, for each node, explicitly represents the relative probabilities of its child nodes. Then it uses a random walk to assign probabilities to words. More formally, each word w can be reached by an appropriate path from the root of the tree. Let $n(w,j)$ be the j^{th} node on the path from the root to w, and let $L(w)$ be the length of this path, so $n(w,1) = root$ and $n(w,L(w)) = w$. Furthermore, for any inner node n, let $ch(n)$ be an arbitrary fixed child of n and let $[\![x]\!]$ be 1 if x is true and -1 otherwise. Then the hierarchical softmax defines $p(w_{Output}|w_{Input})$ as follows:

$$p(w_{Output}|w_{Input}) = \Pi_{j=1}^{L(w)-1} \sigma\Big([\![n(w,j+1) = ch(n(w,j))]\!]\Big) \cdot v'_n(w,j)^T v_{w_{Input}} \quad (3.22)$$

where $\sigma(x) = 1/(1+\exp(-x))$. It can be verified that $\sum_{w=1}^{V} p(w|w_{Input})$ and $\nabla \log p(w_{Output}|w_{Input})$ is proportional to $L(w_{Output})$, which on average is no greater that $\log V$ (Mikolov et al., 2013d).

Negative Sampling

It was introduced by Mikolov et al. (Mikolov et al., 2013d) and used as standard component for both the CBOW and skip-gram models of word2vec. It replaces the softmax – which its gradient is dependant on the summation across all classes – with binary classifiers to prevent expensive and slow training.

The negative sampling is then defined by the objective:

$$\log \sigma({v'_{w_{Output}}}^T v_{w_{Input}}) + \sum_{i=1}^{k} \mathbb{E}_{w_i} \sim P_n(w) \big[\log \sigma(-{v'_{w_i}}^T v_{w_{Input}}) \big] \quad (3.23)$$

It is used to replace every $\log p(w_{Output}|w_{Input})$ term in the skip-gram objective. Hence, the aim is to distinguish the target output word w_{Output} from draws from the noise distribution $P_n(w)$ using logistic regression, where there are k negative samples for each data sample. The noise distribution $P_n(w)$ is empirically defined as the unigram distribution of the words to the $\frac{3}{4}^{th}$ power: $P_n(w) = U(w)^{\frac{3}{4}} / \sum_{i=1}^{V} U(w_i)^{\frac{3}{4}}$ (J. Zhang et al., 2018).

3.3.2 Hyper-parameters of Word2vec

Word embedding methods depend on several hyper-parameters that have crucial impact on the quality of embeddings. For this reason, Mikolov et *a*l. (Mikolov et al., 2013c; Mikolov et al., 2013a) and Pennington et *a*l. (Pennington et al., 2014b)

– the inventors of the popular low-dimensional embedding word2vec and GloVe, respectively – have deeply studied the optimisation of the embedding parameters, mainly the vector dimension and the context size. The performance of the embeddings has been measured based on *word similarity* that uses cosine distance between pairs of word vectors to evaluate the intrinsic quality of such word representations, and *word analogies* that capture fine-grained semantic and syntactic regularities using vector arithmetic. The optimal parameters have been obtained through training on large Wikipedia and Google News corpora. But, no evidence was given for generalisation of these parameters to any other corpus with a general or specific topic and guarantee the performance of embeddings. However, most of the work using word embeddings relies on these parameters as the default ones.

Unlike work that uses default settings, literature on learning embedding hyper-parameters is relatively short (Levy and Goldberg, 2014; Miñarro-Giménez et al., 2015). Levy and Goldberg (Levy and Goldberg, 2014) followed Mikolov et *a*l. (Mikolov et al., 2013e; Mikolov et al., 2013c) and Pennington et *a*l. (Pennington et al., 2014b) and trained their embeddings on general topic using Wikipedia corpus. They basically tested their model with different vector dimensions and different window sizes aiming to study the impact of syntactic contexts – that are derived from automatically produced dependency parse-trees – on detecting functional similarities of cohyponym nature. While Miñarro-Giménez *et al.* (Miñarro-Giménez et al., 2015) trained their word embeddings on a domain-specific corpus of medical data in order to study the ability of word embeddings (word2vec) to capture linguistic regularities in the medical corpora. Similar to the previous work, Miñarro-Giménez *e*t al. trained their word2vec embeddings with different parameter settings, *i*.e., dimensionality of vector space, context size, and different model architectures, *i*.e., CBOW and SG (Mikolov et al., 2013e), and simultaneously compared the relationships identified by word2vec with manually curated information from National Drug File – Reference Terminology ontology as a gold standard using word similarity and word analogies in order to evaluate the effectiveness of word2vec in identifying properties of pharmaceuticals and medical relationships. The obtained results (49% accuracy) revealed the unsuitability of word2vec for applications requiring high precision like medical applications. While this research work seems interesting mainly with its appeal to setting hyper-parameters for domain-specific word embeddings, it does not bring a defined method to efficiently set these parameters.

Leading on from the aforementioned observation, the work presented in this thesis lies within the context of word embedding hyper-parametrisation for domain-specific use. The proposed domain to investigate is the *s*cientific domain and more specifically computer science with *machine learning*, as a case study.

There have been some efforts to integrate word embeddings in the scientific domain (Heffernan and Teufel, 2018b; Lu et al., 2018b; Shu Zhao et al., 2018b) for clustering scientific documents based on their functional structures (Lu et al., 2018b) or for identifying problem-solving patterns in scientific text (Heffernan and Teufel, 2018b) or for paper-reviewer recommendation (Shu Zhao et al., 2018b) or for extracting domain knowledge from rich text (Amin et al., 2020). All the previous research work integrated word embeddings as features for their learning algorithms using either arbitrary or default settings (Mikolov's settings (Mikolov et al., 2013e)). However, none of them has focused on training the embeddings and methodologically setting the hyper-parameters suitable for scientific text.

According to the literature, the work described in Chapter 4 represents the first

attempt to methodologically set word embeddings hyper-parameters in the scientific domain.

3.4 Temporal Word Embeddings

Recent years have witnessed a great interest in *computational linguistics* and more precisely *word embeddings* due to their ability to detect word semantics and meanings, which helps to understand and extract knowledge from human language content. Assuming that human language is evolving throughout time and consequently words are continuously changing meanings, *temporal word embeddings* have been recently proposed to track semantic shifts.

Although the study of *temporal word embeddings* is relatively new, some work has emerged (Kutuzov et al., 2018) on how to leverage word embeddings for time-aware knowledge extraction tasks such as sentiment analysis (Hamilton et al., 2016b; Huang et al., 2017) or temporal information retrieval (Chenliang Li et al., 2017; Rosin et al., 2017). In general, the approaches in previous work can be categorised into two main categories according to (Kutuzov et al., 2018): *linguistic studies* and *event detection approaches*.

Linguistic studies focus on learning and understanding the semantic shifts of human language in general context. As a matter of fact, these studies aim to (i) explore and analyse emerging word meanings and semantic shifts of particular words (Carlo et al., 2019; Y. Kim et al., 2014; Kulkarni et al., 2015) or sentiment words (Hamilton et al., 2016b), (ii) detect temporal correspondence that requires finding different words with semantically similar meanings at different points in time (Szymanski, 2017; Y. Zhang et al., 2016), (iii) identify changes in word usage overtime using word epoch disambiguation (Dubossarsky et al., 2019; Rada and Vivi, 2012), and (iv) reveal statistical laws of semantic evolution (Hamilton et al., 2016a). While *linguistic studies* attempted to trace temporal changes in language semantics in a general context, *event detection approaches* have been proposed to track the 'cultural' semantic shifts that follow real-world events such as tracing armed conflicts (Kutuzov et al., 2017), performing a time-sensitive query expansion for temporal information retrieval (Rosin et al., 2017) or detecting trending concepts behind words (Yao et al., 2018a).

Following this trend, the work presented in this thesis tends to be placed, where both categories will be employed to trace evolving keywords in scientific language in order to detect trending scientific topics and track the dynamics of science. This thesis concentrates on the area of machine learning. According to the literature, the proposed approach represents the first attempt harnessing *temporal word embeddings* in a domain-specific language – scientific language, aiming to perform the computational history of science in the area of machine learning.

3.5 Word Embeddings – Other Techniques

This section presents and describes other word embedding techniques.

3.5.1 GloVe

GloVe, for *Global Vectors*, is a model for distributed word representation that captures the global statistics of the corpus (Pennington et al., 2014a). It has been introduced in 2014. The model is an unsupervised method for learning vector representation

of words from these statistics. It aims to achieve two goals: (i) create word vectors that capture meaning and analogy in the vector space, and (ii) take advantage of the aggregated global word-word co-occurrence statistics instead of only local context window methods (such as skip-gram).

To achieve these goals, the first step is to build a co-occurrence matrix. GloVe also takes local context into account by computing the co-occurrence matrix using a fixed window size. Once the co-occurrence matrix is built, the principle of GloVe is to predict the co-occurrence ratios between two words in a context. Let X refer to the co-occurrence matrix and X_{ij} refer to the i, j^{th} element in X, which is equal to the number of times word j appears in the context of word i. Let $X_i = \sum_k X_{ik}$ be the total number of words that have appeared in the context of i. The relation between the ratios is defined as follows:

$$F(w_i, w_j, \tilde{w}_k) \approx \frac{P_{ij}}{P_{jk}} \tag{3.24}$$

where P_{ij} refers to the probability of the word j appearing in the context of i, and can be computed as $P_{ij} = \frac{\text{number of times j appeared in context of i}}{\text{number of words that appeared in context of i}} = \frac{X_{ij}}{X_i}$, F is some unknown function that takes the embeddings for the words i, k, j as input, and $w \in \mathbb{R}^d$ are word vectors and $\tilde{w} \in \mathbb{R}^d$ are context word vectors, with d the vector dimensionality.

GloVe has been used for different applications such as finding relations between words like synonyms, company-product relations, zip codes and cities, *etc.* It has been also used as a word representation model to detect psychological distress in adults through transcriptions of clinical interviews (Correia et al., 2016).

3.5.2 FastText

FastText is a neural networks -based model for vector representation of words, where each word is represented as a bag of characters n-grams (Bojanowski et al., 2017). The model is created by Facebook's AI Research[1], and has been introduced in 2016. The model architecture is based on the skip-gram model (Mikolov et al., 2013e), and it is considered as an unsupervised method for learning word representations while taking into account morphology. The morphology is modeled by considering subword units, and representing words by a sum of its character n-grams. More formally, given a dictionary of n-grams of size G and a word w, let denote by $\mathcal{G}_w \subset \{1, \ldots, G\}$ the set of n-grams appearing in w. A vector representation z_g is associated to each n-gram g. The word w is then represented by the sum of the vector representations of its n-grams. The scoring function is thus defined as follows:

$$score(w, w_{Output}) = \sum_{g \in \mathcal{G}_w} z_g^T v_{w_{Output}} \tag{3.25}$$

where w_{Output} is a word context and $v_{w_{Output}}$ is its vector representation. This model accordingly shares representations across words, which allows to learn reliable representation for rare words. Instead of explicitly using word order, it uses a bag of n-grams to maintain efficiency without losing on accuracy. It then uses a *hashing trick* (Joulin et al., 2016) to maintain fast and memory efficient mapping of the n-grams into integers in 1 to K. The Fowler-Noll-Vo hashing function[2] has been

[1] https://ai.facebook.com/research/

[2] www.isthe.com/chongo/tech/comp/fnv/

used to hash character sequences by setting K to 2×10^6. Eventually, a word w is represented by its index in the word dictionary G and the set of hashed n-grams it contains.

FastText has been mainly used for text classification. In fact, it has been defined as a fast text classifier (Joulin et al., 2016) used for sentiment analysis (Joulin et al., 2016; Kula et al., 2020) and tag prediction tasks (Joulin et al., 2016). Although it uses shallow neural networks, the obtained results in term of accuracy are on par with deep learning methods, while being much faster (Joulin et al., 2016).

3.5.3 BERT

BERT, which stands for *Bidirectional Encoder Representations from Transformers*, is a transformer-based language representation model, developed by Google and published in 2019 (Devlin et al., 2019). Unlike previous models such as word2vec (Mikolov et al., 2013e) and GloVe (Pennington et al., 2014a) that generate a single word embedding representation for each word in the vocabulary, BERT provides a contextualised embedding by taking into account the context for each occurrence of a given word. This contextualised embedding is reached by pre-training deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. Consequently, the pre-trained BERT model can be fine-tuned with just one additional output layer to build state-of-the-art models for a wide range of natural language understanding tasks, such as question answering and language inference, without substantial task-specific architecture modifications (Devlin et al., 2019).

3.5.4 Discussion

Together with word2vec (Mikolov et al., 2013e), the word embedding techniques described in Section 3.5 – namely GloVe (Pennington et al., 2014a), FastText (Bojanowski et al., 2017) and BERT (Devlin et al., 2019) – aim to provide vector representations for words in order to promote automatic knowledge extraction from unstructured text. This is useful for different NLP tasks such as semantic text analogy, word-sense disambiguation, sentiment analysis, *etc.* Considering that this thesis will tackle the semantic text analogy within the scientific language, therefore any word embedding technique could be applied. However, word2vec (Mikolov et al., 2013e) is chosen to be applied in this thesis for the following reasons:

- Word2vec is the long standing word embedding technique in the area.
- Word2vec has performed better in most cases in the comparative study conducted by Wang *et al.* (Wang et al., 2019).
- With a computing-related language – a language that corresponds to a dataset collected at the German Research Center for Artificial Intelligence (DFKI) – word2vec has performed better than FastText in a comparative study performed by Amin *et al.* (Amin et al., 2018). In other hand, FastText tends to perform better with morphology-related tasks and vocabularies with unknown and rare words such as social media language, while word2vec performs better with semantic tasks, which is the case of the task this thesis gets into.
- BERT provides different vector representations (embeddings) for a single word. These different embeddings for a single scientific keyword do not help

to track the semantic change in similarities between pairs of keywords over time. Consequently, BERT does not help with the aim of this thesis. Furthermore, it has been proven that transformer-based models (BERT) (Devlin et al., 2019) without fine-tuning are usually less useful than plain word2vec (Peters et al., 2019), which is the case of scientific language that does not need a lot of fine-tuning.

- The recently emerging related work (J. He and Chen, 2018; Vahe et al., 2019) that attempted to explore word embeddings within the scientific language have used word2vec to represent the scientific text, which helps to maintain comparative studies if applicable.

Relying on the reasons stated above, word2vec is used in this thesis to represent the scientific language, and its SG neural network architecture is adopted as it consistently proved to be experimentally better than CBOW architecture (Mikolov et al., 2013b).

3.6 Summary

This chapter, in the first place, stated the history of word embeddings. In the second place, it presented the foundations of word embeddings. More specifically, it described vector space semantics including the principles of vector space models and the different similarity measures used in the literature. It also presented the most common distributional semantic models, in addition to machine learning. In the third place, the chapter extensively described the two word2vec models – *continuous bag-of-words* model and *skip-gram* model, in addition to the used computation techniques hierarchical softmax and negative sampling. In the fourth place, this chapter summarised research work on temporal word embeddings. This is justified by the fact that temporal word embeddings will be leveraged in the approaches being described in the next chapters. Finally, in the fifth place, the chapter described other word embedding techniques found in the literature and justifies the choice of word2vec in this thesis.

In this thesis, word2vec is used as a word embedding technique to treat and analyse scientific text. Despite their popularity in overwhelming state of the art performance in semantic similarity and analogy tasks, word embeddings are still treated as black boxes and uniformly use the hyper-parameters without a methodological setting. From this perspective and aiming to provide precise semantic analogies, which are crucial to maintain an accurate computational history of science, the next chapter will address word embedding hyper-parametrisation for domain-specific use, namely the scientific domain. By proposing the *stability of k-nearest neighbors* of word vectors, this thesis aims to methodologically set the hyper-parameters suitable for scientific text.

Chapter 4

Tuning Word2vec Hyper-parameters using k-NN Stability

> "I don't think you can write novels on the road. You need a certain stability."
>
> — Leonard Cohen. (1934–2016)

In the previous chapter, word embeddings have been described, namely *word2vec* – the word embedding technique used throughout this thesis. It has been mentioned in Chapter 2 that word embeddings have been recently used for the representation of scientific text; they enable the generation of semantic analogies. Considering that the semantic analogies are important in the scientific text and their accuracy directly impacts the way the computational history of science is done, the stability of the hyper-parameters of word embeddings is then crucial to be set. This chapter presents the attempt of this thesis to tackle the challenge of word2vec hyper-parametrisation. It describes the proposed methodological approach for tuning word2vec hyper-parameters by using what is coined in this thesis *the stability of k-nearest neighbors* of word vectors. The proposed approach is applied to scientific corpora and more specifically computer science corpora with machine learning adopted as a case study. The proposed approach is tested on a dataset created from the NIPS[1] publications, and evaluated with a curated Association for Computing Machinery (ACM) hierarchy and Wikipedia machine learning outline as gold standards. Both quantitative and qualitative analyses indicate that the proposed approach not only reliably captures interesting patterns like **"unsupervised_learning** *(which is a category of machine learning)* **is to kmeans** *(k-means, which is a specific technique of machine learning)* **as supervised_learning** *(which is a category of machine learning)* **is to knn** *(k-nearest neighbours or k-NN, which is a specific technique of machine learning)***"**[2], but also captures the analogical hierarchy structure of machine learning and consistently outperforms the 61% sate-of-the-art embeddings on syntactic accuracy with 68%.

This chapter is organised as follows. Section 4.1 introduces the research context. Section 4.2 presents the proposed methodology and how the stability of k-nearest neighbors is used to optimise word2vec hyper-parameters. Section 4.3 describes the

[1] Conference on Neural Information Processing Systems

[2] The keywords 'unsupervised_learning", "kmeans", 'supervised_learning" and "knn" are spelled here exactly as they are spelled in the dataset, and as they are learned by word2vec.

NIPS dataset, the analogy dataset created from ACM hierarchy and Wikipedia as gold standards, presents and discusses the results. Finally, the chapter concludes by a summary in Setion 4.4. The work described in this chapter has been published in (Dridi et al., 2018).

4.1 Research Context

The motivation in this chapter is to deeply understand the embedding behavior within scientific corpora, which is quite different to other corpora in terms of word distributions and contexts. For instance, the term *"learning"* appears in the context of education in newspapers corpora; however, *"learning"* appears in a completely different context within computer science. Therefore, word embeddings for scientific text are worth investigating.

There have been some efforts to integrate word embeddings in the scientific domain (Heffernan and Teufel, 2018b; Lu et al., 2018b; Shu Zhao et al., 2018b); however, these efforts do not study learning the hyper-parameters suitable for a scientific text, and instead use either arbitrary or default settings (Mikolov's settings (Mikolov et al., 2013e)).

This thesis aims to fill this gap. Hypothesising that by devising an approach for setting hyper-parameters of word embeddings in the scientific domain, this study adds a deep understanding of the sensitivity of embeddings to hyper-parametrisation. To make the point, this work proposes to use the *stability of k-nearest neighbors* of word vectors as a measure to set the hyper-parameters – mainly *vector dimensionality* and *context size* – of word vector embeddings; moreover, it proposes using common-sense knowledge from the *ACM hierarchy*[3] and *Wikipedia outline of machine learning*[4]. As a result, this work adds breadth to the debate on the strengths of using word embeddings for knowledge extraction from scientific text. According to the literature, the proposed work represents the first attempt to methodically set word embeddings hyper-parameters in a scientific domain.

4.2 Methodology

This study focuses on word2vec hyper-parameter optimisation applied to scientific publications, *i.e.*, how to tune the hyper-parameters that have the largest impact in the prediction performance and what are the adoptable techniques to test the potential of word embeddings for identifying relationships from unstructured scientific text. Accordingly, the *k-NN algorithmic stability* is adopted to investigate the marginal importance of hyper-parameters of skip-gram architecture in a scientific setting. This allows us to identify three hyper-parameters, namely *vocabulary subsampling*, *vector dimensionality* and *context size*, which can significantly affect the embedding performance. In this study, the popular variant of word2vec architecture: *skip-gram* is used as it is consistently yielded superior results comparing to *CBOW* architecture (Mikolov et al., 2013e).

[3] https://dl.acm.org/ccs/ccs_flat.cfm
[4] https://en.wikipedia.org/wiki/Outline_of_machine_learning

4.2.1 The Skip-gram Model

Previous results reported in the literature have shown that skip-gram (Mikolov et al., 2013e) model does not only produce useful word representations, but it is also efficient to train. For this reason, this thesis focuses on it to build the embeddings for scientific text in this study. As described in Section 3.3.1 in Chapter 3, the main idea of skip-gram is to predict the *context* c given a word w. Note that the *context* is a window around w of maximum size L. More formally, each word $w \in W$ and each context $c \in C$ are represented as vectors $\overrightarrow{w} \in \mathbb{R}^d$ and $\overrightarrow{c} \in \mathbb{R}^d$, respectively, where $W = \{w_1, \cdots, w_V\}$ is the words vocabulary, C is the context vocabulary, and d is the embedding dimensionality. Recall that the vectors parameters are latent and need to be learned by maximising a function of products $\overrightarrow{w} \cdot \overrightarrow{c}$.

More specifically, given the word sequence W resulted from the scientific corpus, the objective of skip-gram model is to maximise the average log probability: $L(W) = \frac{1}{V}\sum_{i=1}^{V}\sum_{-l\leq c\leq l, c\neq 0} logProb(w_{i+c}|w_i)$ where l is the context size of a target word. Skip-gram formulates the probability $Prob(w_c|w_i)$ using a softmax function as follows: $Prob(w_c|w_i) = \frac{\exp(\overrightarrow{w_c}\cdot\overrightarrow{w_i})}{\sum_{w_i\in W}\exp(\overrightarrow{w_c}\cdot\overrightarrow{w_i})}$ where $\overrightarrow{w_i}$ and $\overrightarrow{w_c}$ are, respectively, the vector representations of target word w_i and context word w_c, and W is the word vocabulary. In order to make the model efficient for learning, the hierarchical softmax and negative sampling techniques are used following Mikolov et *al.* (Mikolov et al., 2013e).

Word embedding vectors learned with skip-gram can be used for computing word similarities. The similarity of two words w_i and w_j can simply be measured with the inner product of their word vectors, namely $similarity(w_i, w_j) = \overrightarrow{w_i} \cdot \overrightarrow{w_j}$. Recall that *cosine distance* is the measure used to calculate the similarity between embedding vectors $\overrightarrow{w_i}$ and $\overrightarrow{w_j}$ as following:

$$similarity(w_i, w_j) = cosineDistance(\overrightarrow{w_i}, \overrightarrow{w_j}) = \frac{\overrightarrow{w_i} \cdot \overrightarrow{w_j}}{\|\overrightarrow{w_i}\| \cdot \|\overrightarrow{w_j}\|} \tag{4.1}$$

As discussed in Section 4.1, this chapter aims to evaluate the representation capability of word embeddings within scientific text using word similarities as a pivot to stabilise the embedding hyper-parameters.

Skip-gram model uses a target word w to predict the surrounding window of context words. It weights nearby context words more heavily than more distant context words (Mikolov et al., 2013e; Mikolov et al., 2013c). Results of word2vec training are sensitive to parametrisation. To this end, the aim of hyper-parameter optimisation is to find a tuple of hyper-parameters that yields an optimal model minimising the *loss function* for negative samples $(w, \bar{c})$, where $\bar{c}$ does not necessarily appear in the context of w. This *loss function* $\mathscr{L}$ is defined as follows: $\mathscr{L} = -log(\sigma(\overrightarrow{w} \cdot \overrightarrow{c})) - \sum_{k=1}^{n} log(\sigma(-\overrightarrow{w} \cdot \overrightarrow{\bar{c}_k}))$ where σ is the sigmoid function. For each pair (w, c), the skip-gram model forms n negative pairs $(w, \bar{c}_k)_{k\in\{1,\cdots,n\}}$ by sampling words that are more frequent than some threshold θ with a probability: $Prob(c) = \frac{freq(c)-\theta}{freq(c)} - \sqrt{\frac{\theta}{freq(c)}}$ where $freq(c)$ represents the frequency of the word c.

Word2vec has different hyper-parameters, but *sub-sampling* that automatically affects the *corpus size*, *vector dimensionality* and *context window* are described by the developers of word2vec (Mikolov et al., 2013e; Mikolov et al., 2013c) as the most important ones for achieving good results. Consequently, this study focuses on these

hyper-parameters to produce a distributed representation of words in scientific text and evaluate the quality of embeddings in a domain-specific vocabulary.

Sub-sampling: Vocabulary Size

It has been proven in the literature (Mikolov et al., 2013e; Mikolov et al., 2013c; Pennington et al., 2014b) that word2vec embedding quality increases as the corpus size increases. This is expected as longer corpus typically produce better statistics. Following on from this premise, the aim is to investigate the role of vocabulary size in generating accurate embeddings for scientific text.

Unlike previous work that intuitively increments the vocabulary size by combining corpus, in this thesis, it is proposed to use the same corpus trained in two different ways that led to different vocabulary sizes. First, word2vec is trained with unigrams. Second, the model is trained with bigrams by using *word2phrase* – defined by Mikolov et *al.* (Mikolov et al., 2013c) – that learns phrases by progressively joining adjacent pairs of words with an '_' character. Additionally, the frequent words are sub-sampled on two steps, which result into two different vocabulary sizes. Firstly, all stop words and highly frequent academic words appearing in all publications are removed. Secondly, the vocabulary is restricted to words that occur at least 10 times in the scientific corpus. According to Mikolov et *al.* (Mikolov et al., 2013e), this sampling has proven to work well in practice. It accelerates learning and significantly improves the accuracy of the learnt embedding vectors, as it will be shown in Section 4.3.

Vector Dimensionality and Context Window

The optimisation of *vector dimensionality* and *context window* parameters is supposed to be very crucial to achieve accurate results. The quality of embeddings increases with higher dimensionality under the assumption that it increases together with the amount of training data. But after reaching some point, the marginal gain will diminish (Mikolov et al., 2013e).

The window size hyper-parameter corresponds to the span of words in the text that is taken into account, backwards and forwards when iterating through the words during model training. Similarly to the vector dimensionality hyper-parameter, the larger window size results in more topicality. Nevertheless, after a certain point, the marginal gain decreases.

Due to the sensitivity of these hyper-parameters and since hyper- parametrisation is generally known to be data and task dependent (Hutter et al., 2014), optimal hyper-parameter setting is expected to be different for scientific text. Thus, this thesis proposes to study the marginal importance of word2vec hyper-parameters defined above using the *stability of k-nearest neighbors* of word vectors based on word similarities computed with *cosine distance* (Equation (4.1)) between embedding vectors.

***k*-NN Stability for Word2vec Hyper-parametrisation**

Stability is an important aspect of a learning algorithm. It has been widely used in clustering problems (Rinaldo et al., 2012) to assess the quality of a clustering algorithm. Also, it has been applied in high-dimensional regression (Nicolai Meinshausen and Peter Buhlmann, 2010) for training parameter selection. Analogously

and considering that word embeddings present high-dimensional word representations that led to word clusters, the *k-nearest neighbors* is proposed to tune the hyper-parameters of word2vec. *k*-NN is used to cluster similar words based on their cosine similarities.

The basic idea of word embedding stability is the following: embedding quality inevitably depends on tuning hyper-parameters defined previously, namely *vector dimensionality* and *context window*. If accurate values of the tuning hyper-parameters are chosen, then it is expected that the k similar words to a target word w from different embeddings should be similar. Specifically, it is proposed to fix one hyper-parameter, tune the second one by trying different values and training the model for each value. After each training, word similarities are computed and k-nearest neighbors words are defined. The *k-NN stability*, denoted in this thesis by ω, is defined as a simple overlap rate of similar words resulted from two embeddings with different settings.

$$\omega = \frac{\mathcal{S}^{w}_{E_h} \cap \mathcal{S}^{w}_{E_{h'}}}{k} * 100 \tag{4.2}$$

where $\mathcal{S}_{E_h}$ and $\mathcal{S}_{E_{h'}}$ are two sets of similar words to a target word w resulted, respectively, from two embeddings E_h and $E_{h'}$ with different hyper-parameter values. k is the number of nearest neighbors to w given by the cosine similarity. In this study, different values of k have been tested; $k \in \{5, 10, 15, 20\}$, and the stability ω has been computed accordingly. With the settings ($k = 15$) and ($k = 20$), the stability was low. This is justified by the fact that the values 15 and 20 were giving more neighbors, which enables the appearance of arbitrary unrelated topics. However, for the setting $k = 5$, the stability was very high because the returned nearest neighbors words/keywords were syntactically related to the given word/keyword; this is indeed the way word2vec is working. The syntactically related keywords/words refer to plurals, verbs, *etc.* In this thesis, k is then set to 10 because (i) it was qualitatively found that 10 deemed high enough not to go out of the boundaries, and (ii) the stability results were the best with $k = 10$.

4.2.2 Scientific Linguistic Regularities and Analogies

Word2vec embeddings gain their success from their ability to capture syntactic and semantic language regularities. Surprisingly, they characterise each relationship by a relation-specific vector offset (Mikolov et al., 2013a). For example, the famous analogy *"king is to queen as man is to woman"* is encoded in the vector space by the vector arithmetic *"king - man + woman = queen"*. More specifically, the word analogy task aims at answering the question *"man is to woman as king is to — ?"* given the two pairs of words that share a relation (*"man:woman"*, *"king:queen"*), where the identity of the fourth word (*"queen"*) is hidden.

Motivated by this ability of word2vec to identify relationships and capture analogies in textual data without any prior domain knowledge, this ability is evaluated in a domain-specific corpus, namely, scientific publications. The aim is to assess as to what extent word2vec is able to correctly answer analogical questions in scientific text given the complexity of scientific language comparing to natural language.

The scientific word analogy adopted is to query for scientific regularities captured in the vector model through simple vector subtraction and addition. More formally, given two pairs of words $(a : a')$ and $(b : b')$, the aim is to answer the question *(a is to a' as b is to —?)*. Thus, the vector of the hidden word b' will be the vector

$(a' - a + b)$, suggesting that the analogy question can be solved by optimising:

$$\arg\max_{b' \in W}(similarity(b', a' - a + b)) \tag{4.3}$$

where W is the vocabulary and *similarity* is the cosine similarity measure defined in Equation (4.1).

This task is challenging for scientific language as no gold standard is available to evaluate the efficacy of word2vec in identifying linguistic regularities on unstructured scientific text, unlike existing work that use either the gold standard defined by Mikolov et *al.* (Mikolov et al., 2013a) for general natural language tasks or predefined ontologies like NDF-RT ontology[5] for medical domain. To overcome this problem, it is proposed in this thesis to manually curate relationships related to *machine learning* research area from the *ACM hierarchy* and the *Wikipedia machine learning outline*, and define a test set of analogy questions as *semantic questions* following the relation described above. The semantic questions are formed based on the hierarchical tree structure of both the ACM and Wikipedia outline that led to different *"Parent-Children"* relationships. For example, *"supervised_learning"* and *"unsupervised_learning"* are considered two parents for the two children *"classification"* and *"clustering"*, respectively. Accordingly, the analogical question should be *"classification is to supervised_learning as clustering is to —?"*. To correctly answer the question, the model should identify the missing term with a correspondence counted as a correct match by finding the word *"unsupervised_learning"* whose vector representation is closest to the vector *("supervised_learning" - "classification" + "clustering")* according to the cosine similarity. Recall that for the specificity and complexity of scientific language and respecting the interchangeability of scientific terms, instead of using the exact correspondence as the correct match, it is proposed to adopt an approximate correspondence that considers an answer as correct if it belongs to the 10 nearest words given by cosine similarity in order to guarantee the applicability of the generated embeddings in scientific text. This is applied only for *semantic questions*. However, for *syntactic questions*, it is proposed to adopt an exact correspondence. For example, the syntactic question *"classifier is to classifiers as forest is to —?"* is considered correctly answered if and only if the word *"forests"* is the closest to the vector *("classifiers" - "classifier" + "forest")* according to the cosine similarity.

In addition to the *semantic questions* manually curated from ACM and Wikipedia, *syntactic questions* – which are typically analogies about verb tenses/forms and singular/plural forms of nouns – are defined, in order to test the ability of word2vec to capture the syntactic regularities of scientific language.

4.3 Experimental Evaluation

This section describes the machine learning dataset used in the experiments, and presents the obtained results of the proposed methodological approach to tune the hyper-parameters of word2vec. Both quantitative and qualitative analyses are detailed.

[5]National Drug File -Reference Terminology

4.3.1 NIPS Dataset: Description and Vocabulary Setup

To evaluate word embeddings for scientific language, a subset of 2,789 papers in the area of machine learning is used. It has been published in the NIPS between 2012 and 2017. The dataset is publicly available on Kaggle[6] and contains information about papers, authors and the relation papers-authors. The papers database – that defines six features for each paper: the *id*, the *title*, *the event type*, i.e., poster, oral or spotlight presentation, the *PDF name*, the *abstract* and the *paper text* – is used.

The dataset needs to be pre-processed before being used for training the embedding model, since word2vec is very sensitive to vocabulary granularities like punctuation, lowercase, stop words, *etc.*, which have a direct impact on the quality of generated word embeddings. After removing all punctuation and lower-casing the corpus, the pre-processing has the following steps:

i The remove of stop words from the vocabulary using Stanford NLP stop word list[7] enriched by a list of 170 academic stop words that was defined from common academic vocabulary like "*introduction, abstract, conclusion, table, etc.*"

ii The construction of bag of keywords, where keywords are either *unigrams*, or *bigrams* extracted with *word2phrase*. *Word2phrase* is a word2vec package tool that compoundifies n-grams in a text corpus based on a minimum and a maximum frequency (Mikolov et al., 2013d). To compute n-grams, *word2phrase* has to be ran $(n-1)$ times successively. In this case, it was ran only one time as a vocabulary of 1-grams (unigrams) and 2-grams (bigrams) is only needed. The minimum frequency (*min_count* parameter) is set to 10 in order to remove the infrequent words and reduce the model size. Recall that 10 is the by-default value set by (Mikolov et al., 2013d) for the minimum frequency. The *min_count* parameter could be set as a parameter to explore and tuned with different values. However, it does not represent one of the parameters that have impact on the embedding outcomes. For this reason, in this thesis and in the literature in general, this parameter is set to a threshold; all words with total frequency lower than it are ignored. Generally, 10 sounds an acceptable threshold to consider with scientific corpora, as such a keyword that appears less than 10 times in scientific corpora of a certain time period does not represent an important keyword to consider for the analysis. The two settings resulted into different vocabulary sizes $|W_{unigrams}| = 35k$, $|W_{bigrams}| = 96.7k$ and $|W_{downsampled}| = 57k$. The latter ($|W_{downsampled}|$) corresponds to the vocabulary that discards the less frequent words (that appear less than 10 times) in order to accelerate learning.

4.3.2 Word2vec Training Details: Hyper-parameters Optimisation

As described in Section 4.2, *k-NN stability* was used to optimise the word2vec hyper-parameters, namely, *vector dimensionality* and *context window size*.

Vector Dimensionality

k-NN stability ω, with $k = 5$, was used to evaluate the influence of the vector dimensionality hyper-parameter using vector models generated with 20, 30, 50, 100, 150,

[6] https://www.kaggle.com/benhamner/nips-papers/data

[7] github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt

200, 300 and 500 dimensions, skip-gram architecture and three different vocabulary sizes as described in Section 4.3.1.

Table 4.1 shows the results of *k-NN stability* ω values depending on the vector length and vocabulary size. Word2vec model was initially learned with 20-vector dimension. This trained model was used as a seed setting to start computing ω. More specifically, *k-NN stability* ω at 30-vector dimension was computed based on the 20-vector dimension following Equation (4.3) and ,respectively, each ω value is computed based on the results generated by the previous dimensionality setting. The reported results correspond to the stability average of the top 100 frequent words (unigrams and bigrams) in the vocabulary.

It has been clearly seen from the three vocabulary sizes that the stability increases considerably as the dimensionality increases. But after reaching some point, it diminishes or becomes slightly invariant. For instance, for the *unigram vocabulary*, *k*-NN stability reached 67% with 100-dimension vector performing good results comparing to 30 and 50 dimensions. However, it remains basically steady with a slight increase of 1% at 200-dimension. This increase is not remarkable enough to consider 200-dimension better than 100-dimension since a higher dimension of the vectors implies a bigger size of the resulting vector model and more training time. Then, it is noticed that the stability decreases with larger dimensions (300 and 500). Consequently, these results suggest that 100-dimension vector yielded better stability with unigrams vocabulary.

Similarly, *bigrams vocabulary* shows a substantial improvement in *k*-NN stability from 30-dimension to 200-dimension with 68%. Then, it increases slightly with 300 and 500 dimensions with a 1% gain. Hence, for this vocabulary, the optimal dimensionality value can be fixed to 200. Interestingly, the stability results of the *unigram vocabulary* and the *bigram vocabulary* confirm the hypothesis that vector dimensionality and the amount of training data should be increased together to have better results. As a matter of fact, 100 has shown to be the better vector length for *unigram vocabulary* of 35*k* size, while 200 is better for *bigram vocabulary* of 96.7*k* size. On the

TABLE 4.1: *k*-NN stability ω for vector dimensionality optimisation

	D30	D50	D100	D150	D200	D300	D500
unigrams	42%	53%	67%	67%	68%	66%	65%
bigrams	51%	47%	56%	64%	68%	70%	71%
downsampled bigrams	58%	61%	65%	73%	81%	n/a	n/a

other hand, by looking at the stability values at high dimensions (300 and 500), it is noticed that the stability for the *bigrams vocabulary* is higher than that for the *unigrams vocabulary*. This is comprehensibly justified by three facts: (i) this confirms the hypothesis that word2vec model quality increases as corpus size increases (Miñarro-Giménez et al., 2015), (ii) this proves that *n*-gram enhanced skip-gram model performed better than regular skip-gram based only on unigrams, (iii) this confirms the specificity of scientific language and mainly the *Computer Science* area that contains an important number of bigrams like *"machine-learning"*, *"artificial-intelligence"*, *etc.*

Based on these findings, mainly (i) and (ii), the 300 and 500 dimensions were ignored for training the *down-sampled vocabulary*, which is resulted from down-sampling the *bigram vocabulary* as the vocabulary size is smaller (57*k*). It is worthy to note that this down-sampling improved the training speed and most importantly made the *k*-NN stability values more important with 81% at 200-dimension while it

was 68% with *bigram vocabulary* at the same dimension. This was expected as down-sampling makes the word representations significantly more accurate (Mikolov et al., 2013c).

Overall, the *k*-NN stability results obtained through vector dimensionality optimisation show that bigram enhanced skip-gram model performs better with scientific language, 200 is the optimal vector length for the used dataset and the *down-sampled bigram vocabulary* significantly outperforms the two other vocabularies in term of *k*-NN stability and computation time. Note that for all word2vec training rounds with different vocabularies and different vector dimensionalities, the hyper-parameter *window context* was set to 5, the default window size value provided by *gensim*[8].

Window Context

Similarly to the setting followed to optimise vector dimensionality, *k*-NN stability was adopted to find the optimal window size for the used scientific corpus in this study. Building on previous results, the trained vocabulary used is the *down-sampled vocabulary* and the *vector dimensionality* is 200. Word2vec embeddings were generated with skip-gram model and 7 different window sizes ranging from 2 to 8. Word2vec was initially trained with a context window of size 2 as a starting point. Then *k*-NN stability ω was computed, respectively, based on the previous embedding results. Figure 4.1 presents the values of ω that vary context window size. It is clearly seen from the figure that the optimal window size is 6 with a stability of 70% for the used scientific corpus. The obtained results confirm the fact that larger window size results in more topicality and accordingly better accuracy of word representations. However, the marginal gain decreases after a certain point.

Overall, the findings show that the combination of 200-vector dimension with context window of size 6 and down-sampled bigram vocabulary proved to be the best configuration of skip-gram word2vec model. Additionally, the proposed *k-NN stability* – based on word similarity as embedding properties, that is adopted in this study to optimise the word2vec hyper-parameters for scientific text – confirms all hypotheses related to word embeddings supported in the literature and even goes beyond them by giving a standard way to be sure about the stability of results.

4.3.3 Analogy Evaluation

As described in Section 4.2.2, the word analogy task attempts to query for scientific regularities captured in the embedding model – trained with the previously optimised hyper-parameters – through simple vector subtraction and addition.

The created analogy dataset contains 1991 analogical questions, divided into 1871 semantic questions and 120 syntactic questions. The semantic questions were manually curated from ACM hierarchy (406 questions) and Wikipedia outline of *machine learning* (1465 question). The number of relationships generated from Wikipedia are by far greater than the ACM counterpart. This justified by the fact that ACM is more coarse-grained as it covers all the computer science area, while the Wikipedia outline is a fine-grained hierarchy generated specifically for *machine learning* with very detailed algorithms and applications of the area. All questions that contain words that do not exist in the vocabulary were removed from the analogy dataset in order to fairly evaluate embedding analogies. This resulted into 1573

[8] https://radimrehurek.com/gensim/

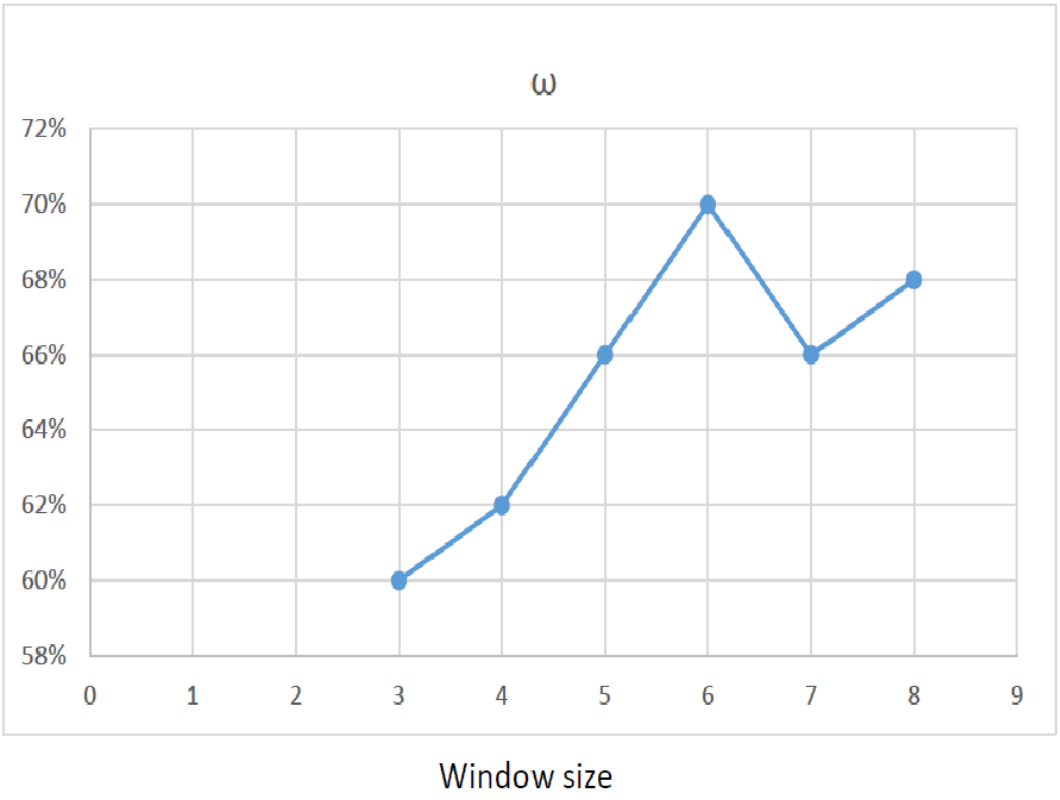


FIGURE 4.1: k-NN stability ω for context window size optimisation

questions (322 ACM questions and 1251 Wikipedia questions). Similarly to semantic questions, syntactic questions were a manually generated subset that is created from the scientific text using typical analogies about verb tenses/forms and singular/plural forms of nouns, in order to test the ability of word2vec to capture the syntactic regularities of scientific language. The number of questions is relatively small due to the aim to only preliminary test the word2vec ability to cover syntactic scientific regularities that do not differ from natural language, while the semantic questions do. That is why more attention was given to these latter. The created analogy dataset is available online for more reproducibility and any further use by researchers[9].

To evaluate the embeddings in capturing linguistic regularities and analogies, both quantitative and qualitative analyses were performed.

Quantitative Analysis

In this analysis, the proposed bigram-enhanced word2vec model – trained with the hyper-parameters experimentally tuned – is empirically evaluated . The goal of these experiments is two-fold. First, it aimed to evaluate whether the hyper-parametrisation method of word2vec is useful for resulting embeddings able to cover linguistic regularities and analogies within scientific text. Second, it aimed to assess whether word embeddings are worth using in domain-specific vocabularies such as the scientific vocabulary.

[9] https://github.com/AmnaKRDB/Machine-Learning-Analogies

To do so, the *accuracy* of word embeddings has been computed to answer the semantic and syntactic questions following the methodology detailed in Section 4.2.2. For semantic questions, 50 out of 322 ACM questions were correctly answered with an accuracy of 15.52% while 75 Wikipedia questions were correct out of a subset of 413 questions from the 1251 questions in the dataset, with an accuracy of 18%. The difference in accuracy between ACM and Wikipedia questions was expected as Wikipedia relationships were more detailed and covered machine learning names of algorithms and applications that widely occur in the vocabulary, while ACM was more coarse-grained. However, the accuracy of both of them is very low. This can be explained by three different reasons. First, the used corpus size is relatively small with only 57k words while it has been shown that word2vec quality increases as corpus size increases. For instance, Mikolov et *al.* (Mikolov et al., 2013c) trained their model on a corpus of 1B words and obtained a semantic accuracy of 61%. Second, the used NIPS dataset is about very recent publications (between 2012 and 2017). So that, the vocabulary is more probably about recent topics and accordingly recent machine learning vocabulary, i.e., names of algorithms and applications might gain more frequencies in the text than the old ones, which in turn would highly affect the word representations, at the time when ACM hierarchy or Wikipedia outline are time-independent and contain generic machine learning vocabulary. Third, the scientific language is complex and does not contain explicit and accurate relationships as natural languages does. For instance, *"accuracy"* and *"error rate"* in the machine learning literature are used in similar contexts, despite having opposite semantics.

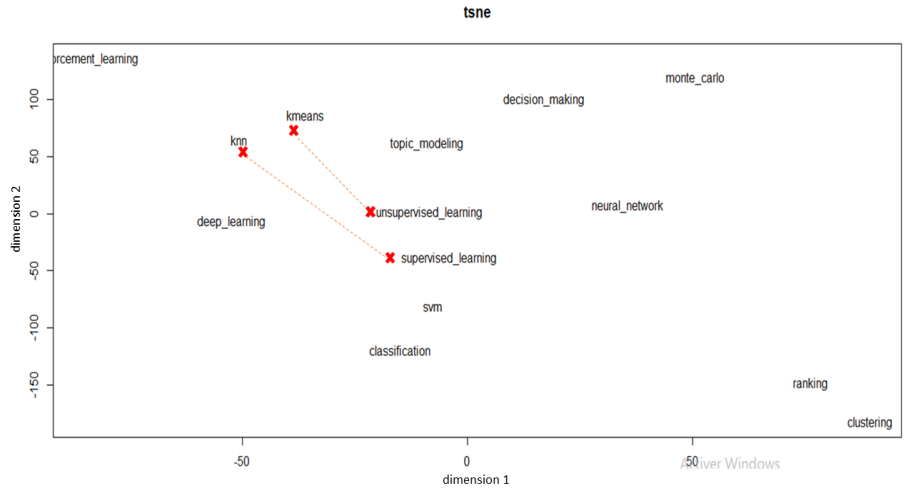
For all these reasons, the semantic accuracy of word embedding within the used scientific corpus is considered modest. But, it is promising as it is interpretable and improvable on one hand. On the other hand, it reveals challenges about scientific word embedding. More specifically, it is worth investigating the convergence and divergence of some *machine learning* algorithms and applications over time, which consistently affects the word representations. Interestingly, it is challenging to find a suitable way to train and evaluate word embeddings in such dynamic vocabularies.

For syntactic questions, the accuracy across the defined 120 questions has been computed. Interestingly, 82 questions out of 120 have been found correctly answered with an accuracy of 68%. This result is interesting despite the small size of the vocabulary. It outperforms the syntactic accuracy of Mikolov et *al.* (Mikolov et al., 2013c), which reached 61% with 1B vocabulary and 300-dimension vector.

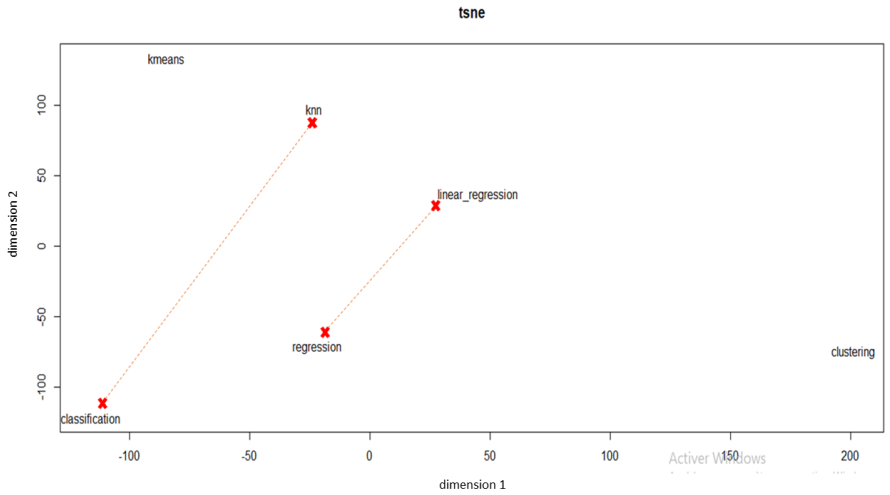
Qualitative Analysis

The learned embeddings revealed interesting patterns in machine learning vocabulary through relation-specific vector offsets. For instance, it captured different semantic relationships mapping machine learning techniques and related algorithms such as r_1:*"unsupervised_learning (which is a category of machine learning) is to kmeans (k-means, which is a specific technique of machine learning) as supervised_learning (which is a category of machine learning) is to knn (k-nearest neighbours or k-NN, which is a specific technique of machine learning)"*, and r_2:*"classification (which is a technique of supervised learning) is to knn (k-nearest neighbours or k-NN, which is a specific technique of supervised learning) as regression (which is a technique of supervised learning) is to linear_regression (which is a specific technique of supervised learning)"*. These patterns are illustrated by plotting word vector representations with *t-distributed stochastic neighbor embedding (t-SNE)* (Maaten and Hinton, 2008) as a qualitative way to evaluate the embeddings

following Yao et *al.* (Yao et al., 2018b). Figure 4.2(a) and Figure 4.2(b) show the t-SNE representations of r_1 and r_2, respectively.



(a) r_1 vector offset



(b) r_2 vector offset

FIGURE 4.2: Vector offsets examples of machine learning semantic relationships

In addition to the t-SNE visualisation used to qualitatively evaluate the accuracy of the embeddings to detect interesting patterns in the scientific text, the capability of the proposed model is proposed to be evaluated in this thesis to capture the hierarchy structure *"Parent-Children"*. To do so, similarities between every word *"parent"* and the corresponding words *"children"* have been computed and compared. The model is considered accurate if the distances are approximately equal. For instance, the distances between the parent *"supervised_learning"* and its children {*"classification"*, *"regression"*, *"ranking"*, *"cost_sensitive"*} are approximately equal with slight differences as presented here, respectively, (0.369; 0.241; 0.173; 0.223) similarly to the parent *"unsupervised_learning"* and its children {*"clustering"*, *"dimensionality_reduction"*, *"topic_modeling"*, *"anomaly_detection"*, *"mixture_modeling"*,

"source_separation"} with approximately similar distances (0.259; 0.307; 0.237; 0.145; 0.145; 0.135; 0.253).

Similarly, the same reasoning to compare the average distances between *"Parent-Children"* has been followed. The model is accurate if the average distance between every parent and its children is similar to others parents' average distances. With respect to the example above, the average distance of the parents *"supervised_learning"* and *"unsupervised_learning"* with their corresponding children has been computed. And interestingly, it has been found that the averages distances are, respectively, equal to 0.25 and 0.22, which proves the accuracy of the embedding to detect granularities of scientific text, not only the semantic relationships but also the hierarchical structure.

4.4 Summary

In the computational history of science, semantic analogies are of crucial. Knowing that word embeddings are able to detect such analogies and being aware that the accuracy of the detected analogies is highly dependant to the hyper-parameters of word embeddings, it is then primordial to effectively tune these hyper-parameters. Despite their popularity in overwhelming state-of-the-art performance in semantic similarity and analogy tasks, word embeddings are still treated as black boxes and uniformly use the hyper-parameters without a methodological setting. From this perspective, this chapter addressed word embedding hyper-parametrisation for domain-specific use, namely the scientific domain. By proposing the stability of k-nearest neighbors of word vectors, the proposed approach was able to methodologically set the hyper-parameters suitable for scientific text. The method has been validated quantitatively and qualitatively on semantic and syntactic analogies curated from ACM and Wikipedia as gold standards and has proved its effectiveness.

The major contributions of this chapter are listed as follows:

- The stability of k-nearest neighbors of word vectors has been proposed as an objective to measure while learning word2vec hyper-parameters.
- The standard skip-gram model has been enhanced by bigrams using *word2phrase* – that attempts to learn phrases by progressively joining adjacent pairs of words with a '_' character – as a method for corpus augmentation.
- An analogy dataset for *machine learning* has been created by manually curating ACM hierarchy and Wikipedia outline of machine learning.
- The approach has been evaluated quantitatively and qualitatively on the NIPS dataset. The embedding detected interesting semantic relations in machine learning such as *"unsupervised_learning (which is a category of machine learning) is to kmeans (k-means, which is a specific technique of machine learning) as supervised_learning (which is a category of machine learning) is to knn (k-nearest neighbours or k-NN, which is a specific technique of machine learning)"*. The obtained results are therefore both promising and insightful.

In this chapter, the hyper-parameters of word2vec suitable for scientific text have been methodologically set, and their effectiveness to detect semantic similarities and syntactic analogies related to scientific language has been proven. In the next chapters, the set hyper-parameters will be applied to train word embeddings in a temporal setting aiming to perform the computational history of science in the area of

machine learning by detecting emerging scientific trends and tracking the dynamics of scientific keywords.

Chapter 5

Hist2Vec: Detecting The Converging Keywords

> "Neither a wise man nor a brave man lies down on the tracks of history to wait for the train of the future to run over him."
>
> — Dwight D. Eisenhower. (1890–1969)

The previous chapter has investigated the impact of word2vec hyper-parameters on the accuracy of the generated analogies in scientific text. A methodological approach has been proposed to tune the hyper-parameters of word2vec in scientific corpora from the domain of *machine learning*. This is an important step toward performing accurate computational history of science. In this chapter, the computational history of science concerns the detection of *converging keywords* that may lead to scientific trends. To this end, word2vec is tuned with the hyper-parameters set in the previous chapter to conduct a fine-grained content analysis of publications from the NIPS conference. This analysis uses *Hist2Vec*, a temporal word embedding approach that represent words with low-dimensional vectors computed by neural networks. The qualitative and quantitative study reported in this chapter reveals the evolution of the prominent machine learning keywords; this evolution supports the popularity of current research topics in the field.

This chapter is organised as follows. Section 5.1 details *Hist2Vec* methodology and how word embeddings are used to detect converging scientific keywords in the area of machine learning. Section 5.2 describes the evaluation process, presents and discusses the obtained results. Finally, Section 5.3 summarises the chapter. The work described in this chapter has been published in (Dridi et al., 2019a).

5.1 Hist2Vec

This chapter introduces *Hist2Vec* – a computational history approach that tracks the rise or the evolution of scientific keywords by detecting the *converging keywords* in the area of machine learning. Accordingly, a *temporal word embedding* technique, namely word2vec (Mikolov et al., 2013d), is adopted to learn word vectors in a temporal fashion, in order to capture words that get geometrically closer, and hence reveal converging keywords. The skip-gram neural network architecture of word2vec is used as it consistently performed better than the continuous bag of words (CBOW) architecture (Mikolov et al., 2013a).

5.1.1 Skip-gram Model

To learn high-quality distributed vector representations, the skip-gram (SG) neural network model was introduced by Mikolov et *al.* (Mikolov et al., 2013b); SG can successfully capture both the semantic and the syntactic word regularities (Mikolov et al., 2013b). The skip-gram model is thoroughly detailed in Chapter 3, Section 3.3.1.

Notation

Let consider the corpora of the NIPS publications collected across time (1987-2015). Formally, $P = (P_1, \ldots, P_T)$, where each $P_{t|t=1,\ldots,T}$ is the corpus of all publications in the t^{th} timespan, and $V = (word_1, \ldots, word_N)$ is the vocabulary that consists of N words present in the corpora P at any point in time; thus, it is likely for some $word_i \in V$ to not appear at all in some P_t. V comprises both emerging and dying words as they typically occur in scientific corpora.

Given this time-tagged scientific corpora, the goal is to find a dense, low-dimensional vector representation $u_{word_i}(t) \in R^d$, $d \ll |V|$ for each word $word_i \in V$ and each timespan $t = 1, \ldots, T$; d is the *dimensionality* of the word vectors.

In the previous chapter, it has been shown that the optimal hyperparameters values are 200 and 6, respectively, for vector dimensionality d and the context window, for the NIPS corpora used in this thesis. Therefore, the skip-gram model is tuned with these hyperparameters in this work. Although the NIPS corpus used in the previous chapter is a smaller corpus (the 6 recent years of the NIPS publications at the time of the experiments) comparing to the one used in this chapter, the optimal hyperparameters values remain valid because (i) the number of years of publications at every timespan (5 years) in this chapter is roughly the same as the corpus in the previous chapter, and (ii) the word2vec model is trained at every timespan.

5.1.2 Temporal Skip-gram Model

To track the dynamism of skip-gram embeddings and measure the accelerations of potential emerging keywords, it is proposed to create a *similarity matrix* $M_{ij}(t)$ of size $|V'| \times |V'|$, $V' \subset V$, for each timespan t that corresponds to the distance metric between two words. Note that V' represent the set of frequent keywords. All distances between two words $word_i$ (w_i) and $word_j$ (w_j) are calculated by the *cosine similarity* between embedding vectors u_{w_i} and u_{w_j} as defined in Equation 4.1, which is redefined in this Chapter by Equation 5.1. Recall that $M_{ij}(t)$ is a symmetric matrix.

$$sim(w_i, w_j) = cosine(u_{w_i}, u_{w_j}) = \frac{u_{w_i} \cdot u_{w_j}}{\|u_{w_i}\| \cdot \|u_{w_j}\|} \tag{5.1}$$

Then, an *acceleration matrix* A_{ij} of size $|V''| \times |V''|$, $V'' = V'_t \cap V'_{t+1}$, is generated. It corresponds to the acceleration between two words w_i and w_j from t to $t+1$. The acceleration between two words w_i and w_j *acceleration* (w_i, w_j) is defined by Equation 5.2 as follows.

$$acceleration(w_i, w_j) = sim(w_i, w_j)^{t+1} - sim(w_i, w_j)^t \tag{5.2}$$

Two keywords are defined as *converging keywords* if their acceleration over two timespans t and $t+1$ is greater than a defined threshold θ. θ is set to the acceleration average over T, $\theta = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|V''|}\sum_{i,j} acceleration(w_i, w_j)$, w_i and w_j are belonging V''.

Table 5.1 summarises the notation used in this chapter.

TABLE 5.1: Table of notations

w_i, w_j, w_v	i^{th}, j^{th} and v^{th} words in the vocabulary
V	the overall vocabulary
$\|V\|, \|V'\|, \|V''\|$	the sizes of the vocabularies V, V′ and V″
nb_v	the set of neighboring words of word w_v.
$u_{w_j}, u_{w_v}, u_{w_i}$	associated word vectors of w_j, w_v and w_i.
$p(w_j \mid w_v)$	the hierarchical softmax of u_{w_j} and u_{w_v}
T	the total number of timespans
t	a specific timespan
P_t	corpus of all publications in the t^{th} timespan
$u_w(t)$	word vector representation of word w at t
d	the embedding dimension
$U(t)$	embedding matrix of all words of size $v \times d$
$M_{ij}(t)$	the similarity matrix
V'	the vocabulary of top-k frequent words in V
A_{ij}	the acceleration matrix
V''	the intersection of top-k words in t and $t+1$
θ	the average acceleration rate over T

5.2 Experiments

To analyse the computational history in the domain of machine learning, the proposed approach *Hist2Vec* is evaluated on a time-stamped text corpora extracted from the NIPS publications. The experiments demonstrate that the proposed approach finds acceleration between trending keywords over time. This allows to track the evolving scientific discovery in the field by following temporal embeddings. The temporal embeddings are used to define the acceleration of various keywords over subsequent timespans in order to detect the fast converging keywords and subsequently the emerging trends.

5.2.1 NIPS Dataset and Preprocessing

The dataset used in this analysis is a set of 5,991 papers published between 1987 and 2015. The data set was first preprocessed. Data preprocessing consists of two steps as described in Section 4.3.1.

To study the temporal evolution of the trends in machine learning by tracking the converging scientific keywords, the NIPS publications between 1987 and 2015 have been divided into six 5-years timespans; however, the last timespan is 4-years long. Therefore, the skip-gram embeddings of the year t' contain a snapshot of the interactions between keywords in the timespan $(t'-4, t')$. For instance, an embedding of the year 2005 describes how the embeddings of keywords developed in the years 2001 to 2005. The length of the timespan is based on the study performed by Hoonlor *et al.* (Hoonlor et al., 2013) on evolving computer science research. Their investigation showed that the average length of the evolutionary chain is 4.5 years. This choice was also tested successfully by Salatino *et al.* (Salatino et al., 2017). The statistics of the dataset are given in Table 5.2.

Table 5.2 shows a positive trend in the evolution of the number of papers per 5-years over the 1987-2015 study period. The average 5-annual growth rate is 22%, rising to 29.71% in the timespan 2007-2011.

TABLE 5.2: Statistics of the NIPS dataset (1987 – 2015)

Timespan	♯ Papers	♯Words	♯Vocabulary
From 1987 to 1991	5, 71	859, 293	10, 823
From 1992 to 1996	729	1, 096, 455	12, 651
From 1997 to 2001	800	1, 301, 492	13, 471
From 2002 to 2006	1023	2, 020, 697	16, 493
From 2007 to 2011	1327	3, 243, 526	21, 074
From 2012 to 2015	1541	4, 002, 513	24, 299

5.2.2 Results and Discussion

The use of temporal word embeddings has been evaluated on the content analysis of machine learning scientific publications and their impact on the evolution of the main streams of machine learning keywords. To do so, the NIPS publications published between 1987 and 2015 and divided into six 5-years timespans, have been used.

The goal of these experiments is two-fold.

1. Evaluate whether the training data with temporal word embedding representations derived from word2vec skip-gram model (where word is not only a unigram, but can also be a bigram) is useful for tracking trends by detecting the converging keywords for the machine learning domain.

2. Study the concordance between the trend analysis method, and the citation-based analysis method, commonly used in the literature.

To generate temporal embeddings, the first step was a text analysis step. For each timespan t, a corpus P_t of all publications published during this time period is created. Then, after removing stop words, term frequency statistics have been performed on unique words of the vocabulary based on two types of bag-of-words: *unigrams* and *bigrams*, in order to study the evolving keywords over time based on their frequencies. Early findings (Lieberman et al., 2007) illustrated that word frequency itself is correlated with the success of the keywords historically. The relation between dynamism and frequency change has been explored in order to gain insights into emerging keywords in the area of machine learning.

By examining the 20 most frequent unigrams and bigrams from the NIPS publications over the six timespans covering a total of 28 years, it was clear that the frequencies of n-grams evolve considerably over time. Interestingly, it was found that the frequencies of some words (unigrams and bigrams) increase by approximately the same rate in specific timespans. For example, it was found that the frequencies of *"neural"* and *"learning"* rose simultaneously between the timespan (1987-1991) and the timespan (1992-1996); this indicates that learning in this time period primarily relied on neural computation. Interestingly, this observation is justified by bigrams. For instance, it has been noticed increasing frequencies of

"neural-networks", *"reinforcement-learning"* and *"machine-learning"* in the next timespans (1997-2001) and (2007-2011).

It seems that as the word usage increases together, these words merge and become emerging keywords. To test the effectiveness of this observation, and assuming that *"deep-learning"* is the emerging trend or keyword in the area of *machine learning* in the last few years while the dataset is limited to only 2015 publications, the attempt was to investigate an intriguing prediction based on the obtained frequencies. Considering that *"deep learning"* is learning based on neural computation, the frequencies of *"learning"*, *"neural"* and *"deep"* have been tracked over time. Recall that *"deep"* and *"deep learning"* do not appear on top-100 words in all timespans. It has been noticed that suddenly the frequency of *"deep"* had a jump in the timespan (2012-2015) that presents the period of emergence of *deep learning*. Figure 5.1 shows how *"deep learning"* as neural learning appears over this 28-year period. The frequency of *"deep"* remained steady, basically null (equal to 19 on the timespan 1997-2001) until 2005, where it started to rise slightly. Then, it rose dramatically to 2913 in the timespan 2012-2015. In this last time period, the frequencies of the three unigrams rose in a parallel way, which justifies their concordance.

Qualitative Results

The results of the analysis performed for this thesis have shown that the proposed approach *Hist2Vec* results in understandable word embedding trajectories on the NIPS corpora. The converging keywords that accelerate significantly to get close over time can be automatically detected .

Figure 5.2 shows t-SNE representations of the six timespans considering bag-of-unigrams (see Appendix A for better visualisation, where the t-SNE representations are resized), while Figure 5.3 shows t-SNE representations (see Appendix B) of the last timespans considering bag-of-bigrams. t-SNE takes the 200 dimensions via word2vec vectors, then reduces them down to 2-dimensional (x,y) coordinate pairs. The idea is to keep similar words close together on the plane, while maximising the distance between dissimilar words (words are unigrams or bigrams). The 2 D t-SNE projection of each unigram's and bigram's temporal embedding across time has been plotted. For visualisation purposes, only the top-100 and top-20 most frequent unigrams and bigrams have been plotted with t-SNE representations, respectively.

Two unigrams of interest in the t-SNE representations have been picked related to unigrams: *"neural"* and *"learning"*. In all visualisations, the embeddings illustrate significant acceleration between the two unigrams. As a matter of fact, their similarity (cosine similarity) is increasing over time. For instance, it increased from 0.0657 in the second timespan (1992-1992) to 0.2235 in the fifth timespan (2007-2011). Table 5.3 shows an increase of 70% in similarity, which suggests that *learning* was increasingly based on *neural computation/networks* and consequently the combination of these unigrams could lead to emerging keywords.

TABLE 5.3: Temporal similarity between "neural" and "learning"

87-91	92-96	97-01	02-06	07-11	12-15
0.1657	0.0657	0.09650	0.1806	**0.2235**	0.1994

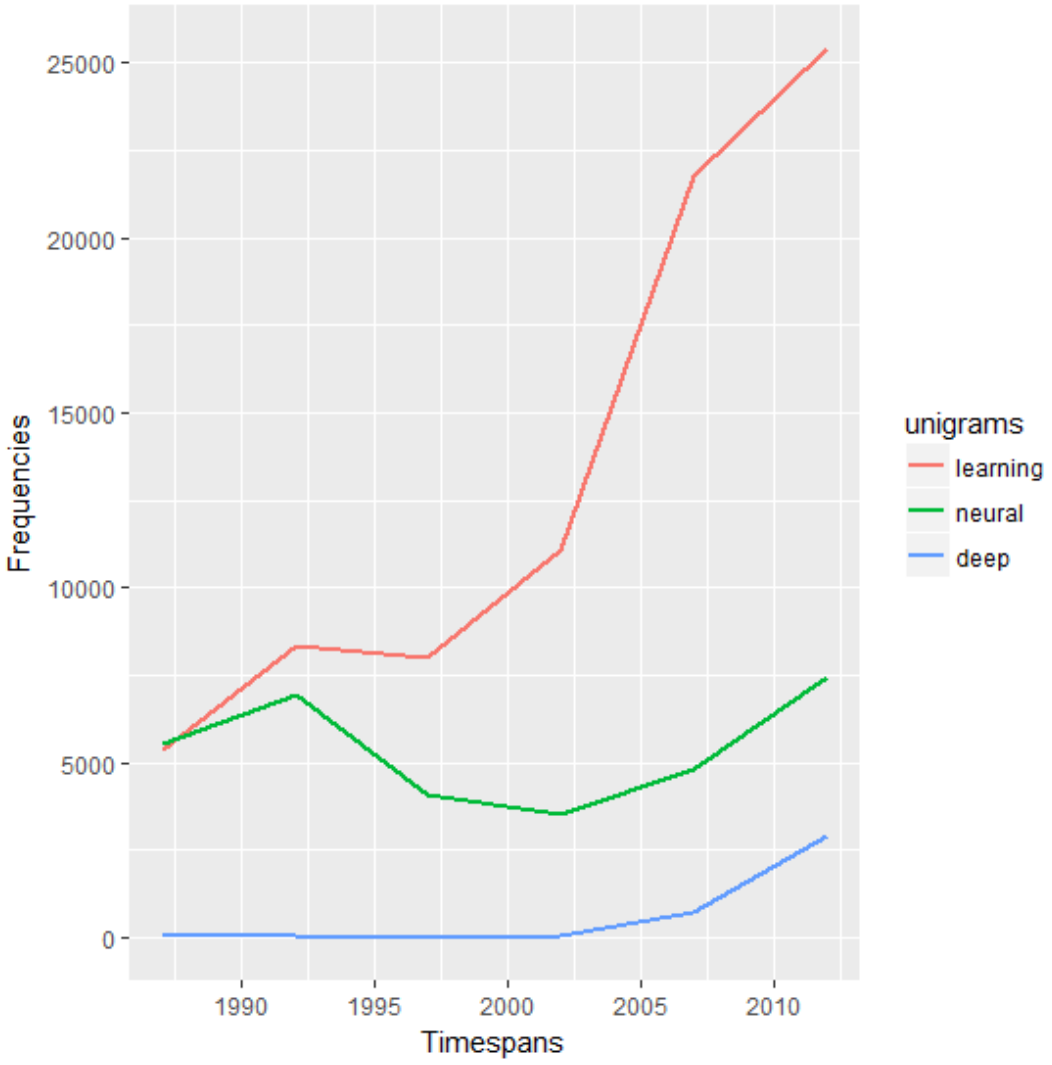
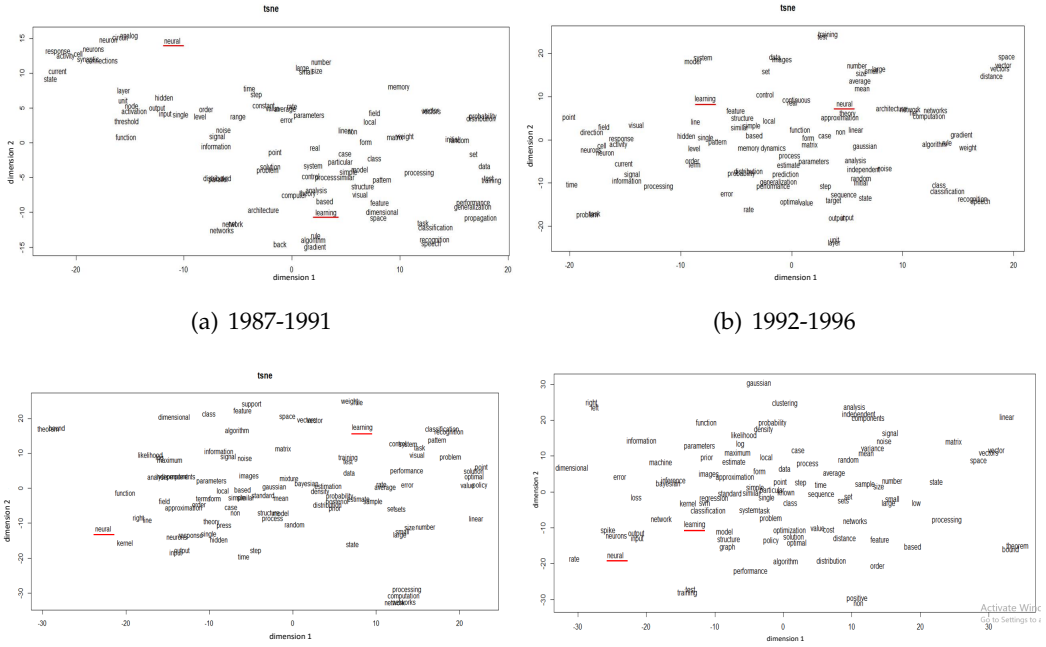


FIGURE 5.1: Frequencies of "deep", "neural" and "learning" over time

Knowing that the unigram *"deep"* is used together with the semantics of *neural computation/networks* and considering that *"deep"* is not represented in top-100 frequent unigrams, the similarity between *"deep"* and *"learning"* has been computed to verify if *"deep"* and *"neural"* get similarly close to *"learning"* over time. Table 5.4 shows that like *"neural"*, *"deep"* gets close to *"learning"* chronologically; in fact, it gets even closer to *"learning"* with a similarity consistently higher than that of *"neural"* over all the timespans. These findings from the temporal embeddings agree with the statistics previously calculated on term frequencies and support the effectiveness of the proposed approach.

TABLE 5.4: Temporal similarity between "deep" and "learning"

87-91	92-96	97-01	02-06	07-11	12-15
0.2285	0.1914	0.1286	0.1885	**0.2569**	0.2458



(a) 1987-1991

(b) 1992-1996

(c) 1997-2001

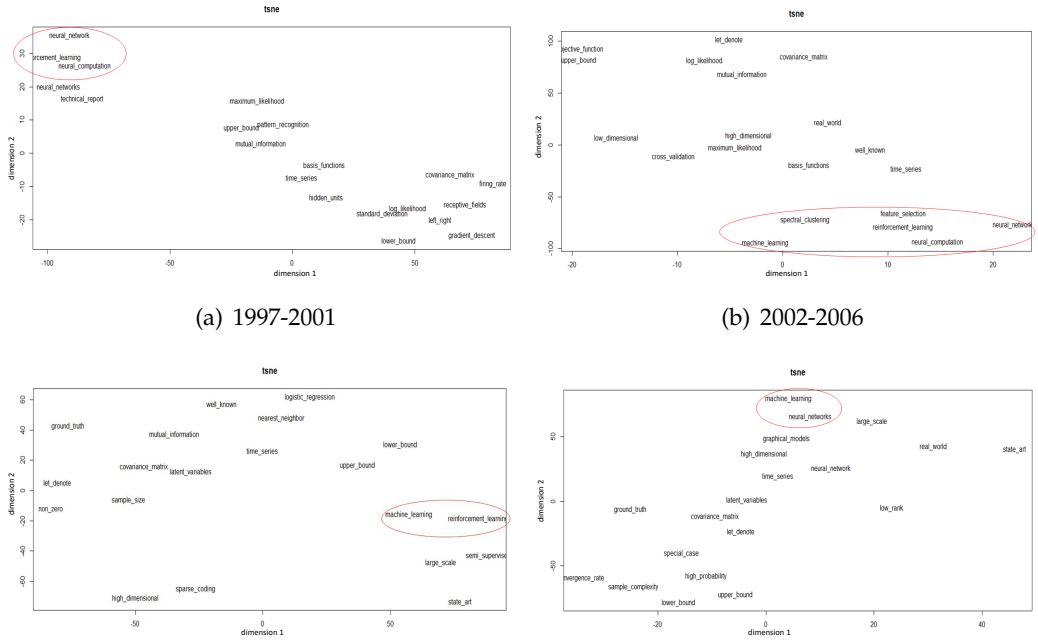
(d) 2002-2006

(e) 2007-2011

(f) 2012-2015

FIGURE 5.2: t-SNE of top 100 unigrams of all timespansthe overlapping keywords – horizontally from left to right, and then vertically from top to bottom – are as follows: in 5.2(a): {“distributed, parallel”, “probability, distribution”, “initial, random”}, in 5.2(b): {“problem, task”, “distribution, probability”, “unit, layer”}, in 5.2(c): {“theorem, bound”, “analysis, independant”, “simple, similar”}, in 5.2(e): {“optimization, normalization”, “dataset, images”, “prediction, classification”, “posterior, distribution”}, and in 5.2(f): {“training, testing”, “model, architecture”, “prediction, classification”, “min, max”, “problem, task”, “performance, accuracy”, “large, small”}

Figure 5.3 shows t-SNE representations of the last four timespans considering bag-of-bigrams. The 2 D t-SNE projection of each bigram's temporal embedding has been plotted across time. For visualisation purposes, only the t-SNE representations of top-20 most frequent bigrams have been plotted.



(a) 1997-2001 (b) 2002-2006

(c) 2007-2011 (d) 2012-2015

FIGURE 5.3: t-SNE of top 20 bigrams of the four timespans between 1997 and 2015

For consistency purpose, t-SNE visualisations for bigrams has been analysed by choosing bigrams similar to the unigrams of interest. The bigrams of interest are: *"neural-networks"*, *"neural-computation"*, *"reinforcement-learning"* and *"machine-learning"*. As only the top-20 bigrams have been plotted, not all the selected bigrams appear in all visualisations. Hence, the focus was only on visualisations of the last four timespans as they mostly contain the bigrams of interest. In Figure 5.3(a) (third timespan: 1997-2001), it is shown the bigram *"reinforcement-learning"* and its neighborhood derived from *"neural"*. i.e. *"neural-network"*, *"neural-networks"* and *"neural-computation"*. This is: (i) semantically significant as *"reinforcement-learning"* by definition is called *neuro-dynamic programming* and needs incremental *"neural networks"*; and (ii) proved by similarity as the latter reaches its peak by 0.9998 during this time period. Likewise, the similarity between *"machine-learning"* and *"neural-networks"* peaks at almost the same value of 0.9997 while *"machine-learning"* is not represented in the figure that shows only the 20 most frequent bigrams. This also indicates that *"reinforcement-learning"* was used as *"machine-learning"* during this time period; in fact, the similarity between *"reinforcement-learning"* and *"machine-learning"* is equal to 0.9994.

One timespan later (2002-2006) (Figure 5.3(b)), *"machine-learning"* appears and its similarity to *"neural-networks"* drops significantly to 0.8686. This shows that *"machine learning"* started to flourish towards the end of 1990s as an independent topic while *"reinforcement learning"* remained linked to *"neural computation/networks"*.

In the fifth timespan (2007-2011) (Figure 5.3(c)) *"neural-networks/computation"* does not appear in the top-20 frequent bigrams. However, this timespan highlights the re-approximation between *"machine learning"* and *"reinforcement learning"* that incorporates *"neural networks"*. This is insightful as it shows how *"machine learning"* is increasingly related to *"neural networks/computation"*.

The last timespan (2012-2015) (Figure 5.3(d)) also shows that *"machine learning"* is geometrically very close to *"neural networks"* that re-appeared, while *"reinforcement learning"* disappeared from the top-20 bigrams. This shows that possibly *"machine learning"* increasingly implies *"neural networks"* just as *"reinforcement learning"* did earlier.

Based on the findings of bigrams embedding and knowing that *"deep-learning"* was the emerging keyword in the last few years, computed the similarity between *"machine-learning"* and *"deep-learning"* has been computed and it has been found that it is equal to 0.8716 while *"deep-learning"* does not exist in previous timespan-vocabularies. Consequently, it can be assumed that *"deep-learning"* is the keyword that emerged from the convergence between *"machine-learning"* and *"neural-networks"*.

The qualitative and quantitative analyses on both unigrams and bigrams show that the learned temporal embeddings reveal interesting patterns in the similarity between potentially emerging keywords over time. To prove that, *similarity matrices* have been created as described in Section 5.1.2 for the top-20 frequent and overlapped bigrams and only a couple of unigrams {*"neural"*, *"learning"*} in order to be consistent to the unigrams and bigrams of interest previously picked. The similarity matrices contain the cosine similarity between the embedding vectors of every pair of keywords.

After creating the similarity matrices, the *acceleration matrix* has been generated as described in Section 5.1.2. The average acceleration θ has been computed, which corresponds to the average over all the selected keywords. θ was negative and equal to -0.0656. Overall, almost all accelerations are negative but some of them were speeding up. For instance, the couple of bigrams {*"neural-networks"*, *"reinforcement-learning"*} have an acceleration of -0.011, which is much greater than the average θ. Interestingly, it has been found that the acceleration of the couple of bigrams {*"neural-networks"*, *"machine-learning"*} is positive and equal to 0.0094. Respectively, the acceleration of the couple of unigrams {*"neural"*, *"learning"*} is positive and has a value of 0.02. Both of them have an acceleration much greater than the average θ. These findings support the previous ones and show that neural-based learning was speeding up over time. Similar to previous investigations about the emergence of *"deep learning"*, the acceleration of two unigrams {*"deep"*, *"learning"*} and two bigrams {*"neural-network"*, *"deep-learning"*} has been computed. Their values are, respectively, 0.0034 and 0.1649, showing a substantial speed up over the average θ.

Quantitative Results

In order to test its effectiveness in detecting emerging keywords in the area of machine learning, *Hist2Vec* has been validated with the citation counting approach, which is widely used in the literature and provides a snapshot of a fast-growing field. The objective is to check the extent to which citation analysis supports the findings of *Hist2Vec*. To do so, academic citations have been retrieved of all the NIPS

publications in the used dataset (1987 to 2015) using the *Public or Perish* software [1] that uses Google Scholar [2] to obtain the raw citations.

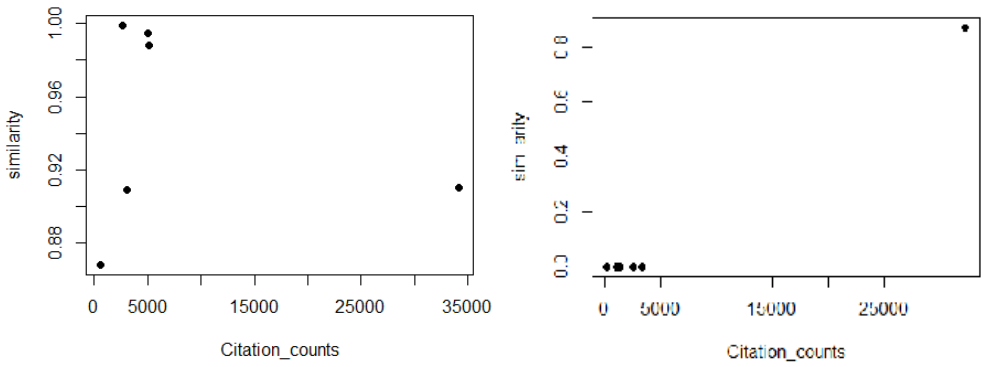
For consistency purpose, the citation counts of publications was tracked with previously selected frequent keywords (the keywords of interest already picked in the qualitative analysis) over time such that in each timespan the citation counts of the publications that used the picked keywords in their titles have been considered; assuming that the title plays a pivotal role in communicating research.

A comparison has been performed between the acceleration of citation counts of publications mentioning the keywords of interest in their titles with the acceleration of similarities of these pairs of keywords over all timespans. *Spearman's correlation coefficient* ρ has been used to measure the strength and direction of association between these two variables. ρ is defined by Equation 5.3 as follows:

$$\rho = \frac{\sum_s (x_s - \bar{x})(y_s - \bar{y})}{\sqrt{\sum_s (x_s - \bar{x})^2 \sum_s (y_s - \bar{y})^2}} \tag{5.3}$$

where s is the paired score $(citation\_count, similarity)$, x and y correspond to the citations counts and similarity values, $\bar{x}$ and $\bar{y}$ correspond, respectively, to the mean of citations counts and the mean of similarity values.

Spearman's correlation coefficient has been computed for all the pairs of picked keywords. Interestingly, it has been found that 100% of cases have a positive correlation with an average of 0.422. 67% of these correlations are strong with ρ coefficient greater than 0.6. Figure 5.4(a) and Figure 5.4(b) show the relationships between



(a) ML – NN (b) NN – DL

FIGURE 5.4: Spearman's coefficient plot of (ML – NN) and (NN – DL)

the citation counts and the similarities of the keywords of interest *"machine-learning – neural-networks (ML – NN)"* and *"neural-networks – deep-learning (NN – DL)"*, respectively. Figure 5.4(a) has Spearman's correlation coefficient ρ equal to 0.2. If the last point was not considered, where the similarity between *"machine-learning"* and *"neural-networks"* dropped in the timespan (2002-2006), ρ coefficient is much higher and equal to 0.9. This observation could be justified by the fact that *"machine learning"* started to flourish towards the end of 1990s as an independent topic, which justifies the decrease in its similarity with *"neural-networks"*. Overall, this new finding confirms the previous findings stating that *learning* was correlated to *neural networks* over time. Figure 5.4(b) has ρ coefficient equal to 0.654. This result

[1] https://harzing.com/resources/publish-or-perish
[2] https://scholar.google.com/

perfectly matches with the previous findings, where the citation count was slightly small in the first four timespans. Then, suddenly it rose dramatically to reach 3,223 in the last timespan (2012-2015). A significant rise of these citation counts is clearly seen, which goes with the increase in the similarity and the acceleration previously detailed, and shows that *"learning"* was increasingly relying on *"neural networks"*. The emerging keyword *"deep-learning"* goes in parallel with the keywords of interest *"neural-networks" and "machine-learning"* and dramatically increased in the last timespan, which supports the assumption that *"deep-learning"* is now the trend.

These findings resulting from citation counts support the effectiveness of the approach based on temporal word embeddings in detecting emerging keywords in the domain of machine learning.

5.3 Summary

In this chapter, *Hist2Vec* has been offered to perform a computational history of science through the detection of *converging keywords* that may lead to new scientific keywords/trends. *Hist2Vec* has been applied to the NIPS publications to produce insights about the field of *machine learning* and track the evolution of new trends.

This work addressed this challenge in an innovative way by bringing together qualitative and quantitative analyses of the NIPS publications during the time period 1987-2015. Both analyses drilled into the paper content by computing and visualising temporal keyword embeddings over six 5-years timespans. The similarity between keywords has been explored by computing the similarity between the embedding vectors in order to create a similarity matrix of frequent keywords. Then, based on this matrix an acceleration matrix has been created, which reports the acceleration between pairs of keywords over time in order to capture the converging keywords that may result in a trending keyword. The results were able to detect that *"deep-learning"* was the convergence between *"machine-learning"* and *"neural-networks"*. *Hist2Vec* has been validated against citation count analysis, and its effectiveness has been demonstrated.

In this chapter, the approach *Hist2Vec* detected the *converging keywords* that may result in trending keywords by computing the acceleration of similarities between keywords over successive timespans. In the next chapter, the rankings of similarities will be adopted and the ascents in ranking over different timespans will be computed to detect the *contextualising keywords* – the keywords to start to occur together in the same context – that may lead to trending keywords. In addition, instead of fixed timespans, the history impact will be explored by adopting both incremental and sliding timespans.

Chapter 6

Leap2Trend: Detecting the Contextualising Keywords

> "I'm an inventor. I became interested in long-term trends because an invention has to make sense in the world in which it is finished, not the world in which it is started."
>
> — Ray Kurzweil. (1948—)

In the previous chapter, the computational history of science has been performed by detecting scientific trends, which have been defined as the *converging keywords* over time. To this end, *Hist2Vec* has been proposed to track the similarity acceleration between pairs of keywords over successive timespans. This chapter continues with the detection of emerging scientific trends. However, it follows a different path to find them. Going beyond the accelerating similarities of pairs of keywords that lead to converging keywords, this chapter studies the dynamics of similarities between pairs of keywords, their rankings and respective uprankings (ascents) over time in order to detect the *contextualising keywords*. This chapter introduces *Leap2Trend*, a novel approach to early detection of research trends. *Leap2Trend* relies on temporal word embeddings (*word2vec*). *Leap2Trend* is applied to two scientific corpora, where machine learning is applied, and it is evaluated against two gold standards *Google Trends hits* and *Google Scholar citations*.

This chapter is organised as follows. Section 6.1 introduces *Leap2Trend* and details its different stages. Section 6.2 describes the used datasets, presents the gold standards, reports and discusses the experimental results. Finally, Section 6.3 summarises the chapter. The work described in this chapter has been published in (Dridi et al., 2019b).

6.1 Leap2Trend

This chapter presents *Leap2Trend*, which is a novel approach for an effective and early detection of emerging scientific trends; defined in this chapter as *contextualising keywords*. *Leap2Trend* follows a fine-grained content analysis approach that digs into textual content of research papers and grasps semantics by applying temporal a word embedding technique, namely word2vec (Mikolov et al., 2013d). Accordingly, temporal word embeddings are adopted. *Leap2Trend* learns these temporal embeddings and tracks the dynamics of pairs of keywords over time in order to capture the fast contextualising keywords, which could led to emerging scientific trends.

The workflow of *Leap2Trend* is depicted in Figure 6.1 and it follows four stages:

i **Data preprocessing.** This stage is conducted to preprocess and clean up data taking into account the specificity of scientific language. It leads to a bag of keywords, where a keyword is either a unigram or a bigram.

ii **Word embeddings.** In this stage, word2vec embedding model is applied with its *skip-gram* architecture (Mikolov et al., 2013e) to learn the distributed vector representations of keywords over time. This stage is repeated for each corpus P_t, $t = \{1, \ldots, T\}$ that corresponds to the corpus of all research papers in the t^{th} timespan.

iii **Similarity computation.** After generating the vector representation of keywords, a *similarity matrix* is created. It corresponds to the cosine similarity between embedding vectors of pairs of keywords. Respectively to the previous stage, this stage is also repeated at each timespan $t = \{1, \ldots, T\}$.

iv **Post-processing.** First, this stage takes as input the previously computed similarity matrix and returns a *ranking matrix* at each timespan t. Then, after defining all ranking matrices corresponding to the T timespans, the identification of pairs of keywords with ascents in their ranking over time is performed. This step is called *rank ascent identification*, which represents the key of the identification of emerging scientific keywords/trends.

In the next sections, the functionalities of these stages are detailed.

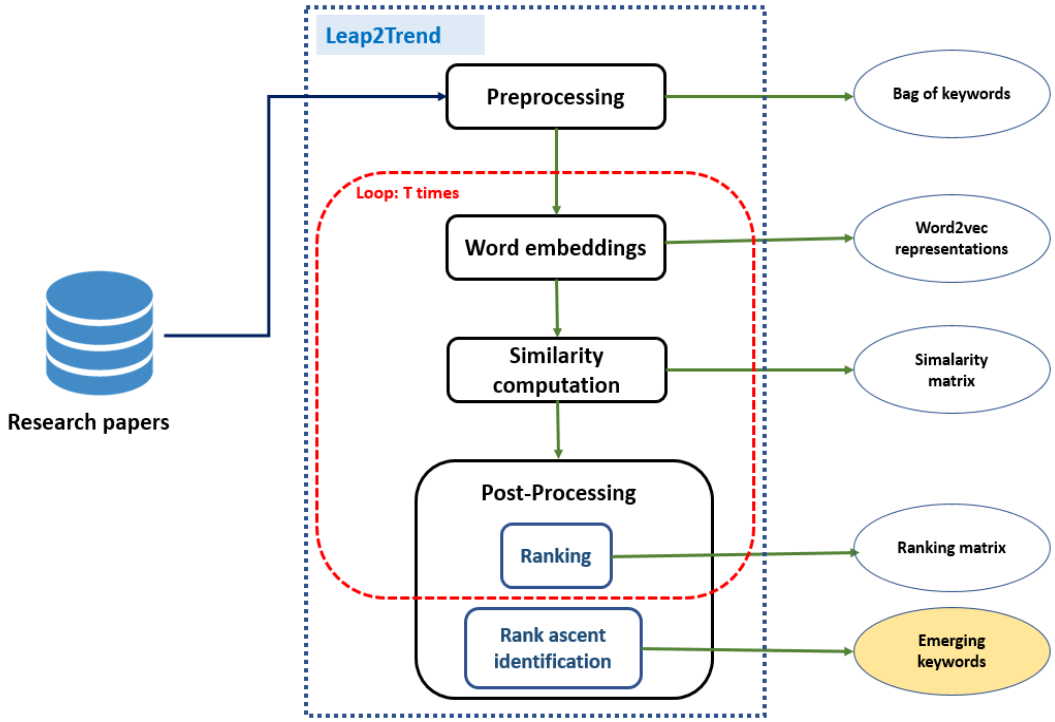


FIGURE 6.1: Workflow of Leap2Trend

6.1.1 Data Preprocessing

This section describes the data preprocessing stages, which are performed following both language and time levels.

Language-based Preprocessing

In order to learn high-quality distributed vector representations of keywords in the scientific text, the first step consists to clean data and take into consideration the specificity of scientific language. For instance, bigrams are commonly used in the scientific language such as *"machine learning"* and *"artificial intelligence"* in the computer science area or *"transfer learning"* and *"breast cancer"* in the bioinformatics area. To do so, two steps are followed as described in Section 4.3.1.

Time-based Preprocessing

After performing a language-based data preprocessing stage, a time-based data processing step is done. It aims to divide the scientific corpora P into T timespans denoted by $P = (P_1, \ldots, P_T)$, where each P_t, $t = \{1, \ldots, T\}$ is the corpus of all research papers in the t^{th} timespan. This step is important to fulfill the temporality of the task of scientific trend detection and track the evolving keywords over time. To this end, a dynamic data integration of corpora is adopted rather than using static time windows. The time-based preprocessing stage has two different temporal paradigms: *incremental windows* and *sliding windows*.

Incremental Windows. Each window or timespan t represents a sequence of time stamped corpora P_t, $t = \{1, \ldots, T\}$ gradually created following a 1-year annual basis. Therefore, the corpus of the window $t'_{1 \leq t' \leq t}$ will contain the aggregated corpora of the timespan $(1, t')$ as illustrated in Figure 6.2. For instance, if the scientific corpora dated from 2000 to 2018, the corpus of the window 2008 will contain all corpora between 2000 and 2008. The corpus of the last window T contains all corpora from window 1 to window T.

The choice of the incremental paradigm is based on the normal flow of scientific venues such as conferences and journals, which are annually publishing new papers. 1-year window length is adopted for the corpus increment in order to keep the study as fine-grained as possible by following a tight track of keywords movement and trend emergence.

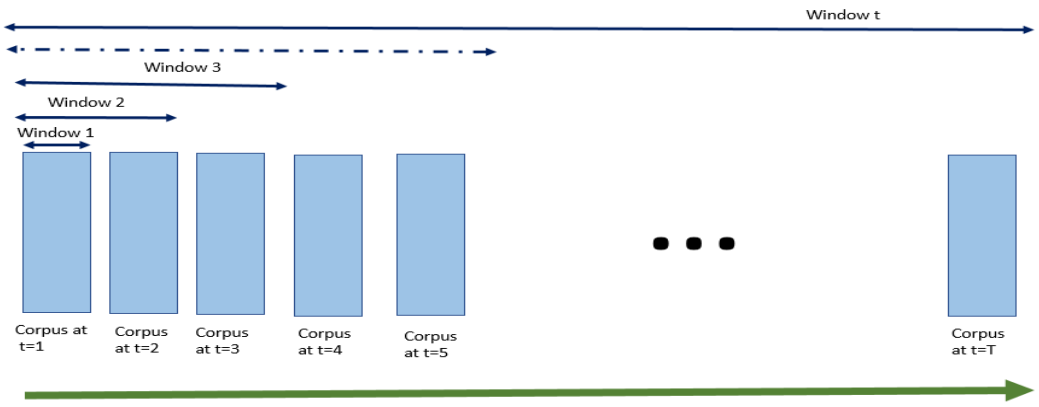


FIGURE 6.2: Incremental windows

Sliding windows Each window t represents a sequence of three time stamped corpora. The corpus of the window t will contain the corpora of the timespan

$(t-1, t+1)$ as shown in Figure 6.3. For instance, the corpus of the window 2008 will contain the corpora between 2007 and 2009.

The choice of the timespan length is based on the study performed by Anderson *et al.* (Ashton et al., 2012) on evolving scientific topics. Their investigations showed that the interval of three years was successful to track the flow of scientific corpora.

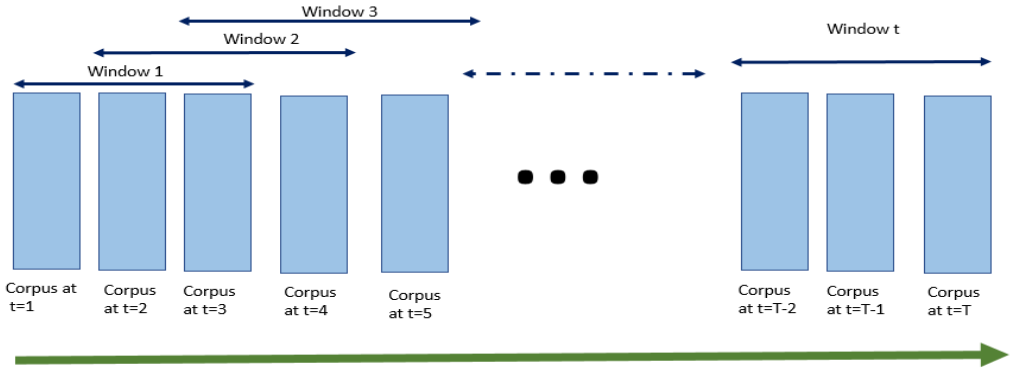


FIGURE 6.3: Sliding windows

6.1.2 Word Embeddings

This study introduces a temporal word embedding approach based that tracks emerging scientific keywords at an early stage by detecting the contextualising keywords that capture the evolution and the movement of pairs of keywords over time. Accordingly, a temporal word embedding technique is adopted to learn word vectors in a temporal fashion. To do so, the skip-gram architecture of word2vec model is used as it consistently proved to be experimentally better than CBOW architecture (Mikolov et al., 2013b).

Skip-gram Model

Skip-gram model has been introduced by Mikolov et *al.* (Mikolov et al., 2013b) for learning high-quality distributed vector representations. The main idea of *skip-gram* is to predict the *context* given a word w_i. Note that the *context* is a window around w_i of maximum size L that represents the span of words in the text, which is taken into account both backwards and forwards when iterating through the words during model training. Skip-gram model is detailed in Chapter 3, Section 3.3.1.

Notation. Let consider corpora of research papers collected across time. Formally, it is denoted by $P = (P_1, P_2, \ldots, P_T)$ the corpora, where each P_t is the corpus of all papers in the t^{th} timespan. Denote $\mathcal{V} = (w_1, w_2 \ldots, w_V)$ the vocabulary that consists of V words present in the corpora P. It is possible that some $w_i \in \mathcal{V}$ not appear at all in some P_t. This comprises emerging keywords and dying keywords that are typical for scientific corpora. Let V_t denote the vocabulary that corresponds to P_t and $|V_t|$ denote the corresponding vocabulary size used in training word embeddings at the t^{th} timespan.

Given this time-tagged scientific corpora, the goal is to find a dense, low-dimensional vector representation $u^t_{w_i} \in \mathbb{R}^N$, $N \ll V_t$ for each word $w_i \in V_t$ at each timespan $t = \{1, \ldots, T\}$. N is the *dimensionality* of word vectors that corresponds to the length of the vector representations of words. Let $\mathscr{W}$ denote the matrix of size $V_t \times N$ that represents the input to hidden layer connections with each row representing a vocabulary word $w_{i,i=1,\ldots,V_t}$, and $\mathscr{W}'$ the matrix of size $N \times V_t$ that describes the connections from the hidden layer to the output layer with each column of $\mathscr{W}'$ representing a word w_i from V_t.

Model. Given the vocabulary of size V_t at timespan t, word embedding vectors of size N are learned. The skip-gram model learns to predict one context word w_j (output) using one target word (input) w_i at a time as following:

- The input word w_i and the output word w_j are one-hot encoded into binary vectors x and y of size V_t.
- The multiplication of the binary vector x and the word embedding matrix $\mathscr{W}$ of size $V_t \times N$ gives the embedding vector of the input word w_i; the i-th row of the matrix $\mathscr{W}$.
- The hidden layer represents the resulting embedding vector of dimension N.
- The multiplication of the hidden layer and the word context matrix $\mathscr{W}'$ of size $N \times V_t$ produces the output one-hot encoded vector y.
- The final output layer applies *softmax function* (Mikolov et al., 2013d) to compute the probability of predicting the output word w_O given the input word w_I, and therefore:

$$p(w_O|w_I) = \frac{exp({v'}^T_{w_O} v_{w_I})}{\sum_{w=1}^{W} exp({v'}^T_w v_{w_I})} \tag{6.1}$$

  where v_w and ${v'}_w$ are the input and output vector representations of w that correspond to x and y in this case, and W is the number of words in the vocabulary that corresponds to V_t in this case.
- The output context matrix $\mathscr{W}'$ encodes the meanings of words as context.

Temporal Word Embeddings

In order to study the dynamics of the skip-gram model and track the movement of potential emerging keywords; defined in this chapter as contextualising keywords, it is proposed to learn word embeddings in a temporal fashion. To do so, the skip-gram model is trained on the data resulting from the time-based preprocessing stage described in Section 6.1.1. Therefore, two training paradigms are proposed with respect to the generated corpora, namely *incremental embedding* for the incremental windows and *sliding embedding* for the sliding windows.

Incremental embedding. The incremental embedding goes through the corpora P to update word embeddings incrementally with the annual basis corpus augmentation. To do so, two different embeddings are proposed. The first embedding aims to retrain the skip-gram model from scratch and perform a fresh model termed as *fresh embedding* in this thesis. The second embedding, termed as *updated embedding*, reads the training data word by word to incrementally

update the word frequency distribution and the noise distribution while performing stochastic gradient descent (Kaji and Kobayashi, 2017). Figure 6.4 illustrates the proposed incremental embedding model.

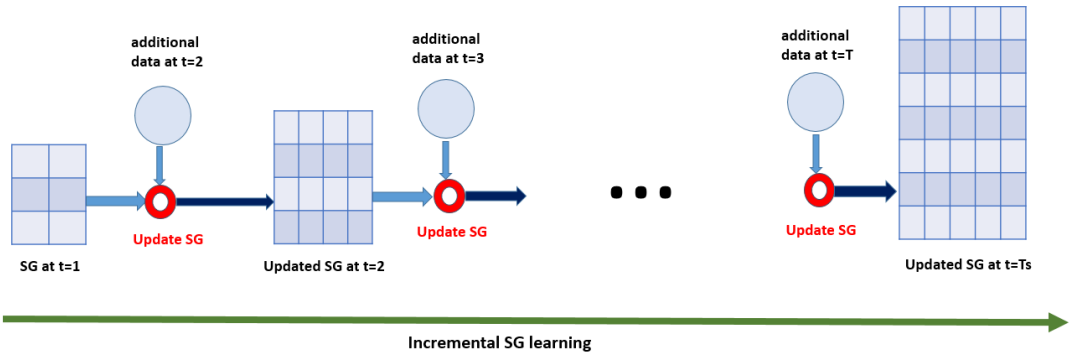


FIGURE 6.4: The incremental embedding model

Sliding embedding. At every timespan t, the sliding embedding considers as input the corpora in the window $(t-1, t+1)$ and trains skip-gram model after creating a new vocabulary $V_t \subseteq \mathcal{V}$ corresponding to the actual window. $\mathcal{V}$ may therefore vary as the window is progressing over time.

After selecting the first three corpora in the window starting from $t-1$ as mentioned in Section 6.1.1, the next corpora is selected from the window starting from t. The process is repeated iteratively until all P_t corpora are trained.

6.1.3 Similarity Computation

At this stage, *Leap2Trend* creates a *similarity matrix* $M^t_{i,j}$ of $|v| \times |v|$, $v \subseteq \mathcal{V}$ for each timespan t, respectively, for both temporal training paradigms of skip-gram model (incremental and sliding). Note that $|v|$ is the number of the most frequent keywords used in the similarity computation across all corpora. It is worth noting that the same keywords have been used over all timespans. The similarity matrix $M^t_{i,j}$ corresponds to the similarity metric between two keywords belonging to v. All distances between two keywords w_i and w_j are calculated using *cosine similarity* between embedding vectors u_{w_i} and u_{w_j} as defined by Equation 6.2. Recall that ${M_{i,j}}^t$ is a symmetric matrix.

$$similarity(w_i, w_j) = cosine(u_{w_i}, u_{w_j}) = \frac{u_{w_i} \cdot u_{w_j}}{\|u_{w_i}\| \|u_{w_j}\|} \tag{6.2}$$

For efficiency purposes, the entries of the similarity matrix ${M_{ij}}^t$ correspond only to a subset of keywords that represent top-k keywords. More details on the selection of keywords will be provided in Section 6.2.

6.1.4 Post-processing

After computing all similarity matrices corresponding to all T timespans, *Leap2Trend* proceeds with the ranking of the similarities of pairs of keywords in each matrix. The resulting ranked matrices are then used to identify the pairs of keywords having significant ascents in their ranking over time. These keywords are defined as contextualising keywords and they are potentially considered as emerging trends

due to their frequency of co-occurrence in the same context. This step is termed as *rank ascent identification*.

Ranking

Given a similarity matrix $M^t_{i,j}$ of size $|v| \times |v|_{v \subseteq \mathcal{V}}$; that corresponds to the similarity values of a set v of keywords at a timespan t, the aim is to rank this matrix in order to define the set of contextualising keywords, which correspond to the pairs of keywords that start to frequently co-occur in the same context at this time period.

The ranking of $M^t_{i,j}$ is defined as the ranking of its entries that correspond to the similarities of pairs of keywords. To speed up the rank calculation and considering that $M^t_{i,j}$ is an symmetric matrix, only the upper triangular part of the matrix is considered, which corresponds to the similarity values above the main diagonal. Hence, ranking the matrix $M^t_{i,j}$ corresponds to the ranking of the upper triangular part. Algorithm 1 highlights the steps of the ranking process.

Algorithm 1: Ranking Similarity Matrix

input : similarity matrix $M^t_{i,j}$
output: ranked $M''^t_{i,j}$

```
1  rank ← 0;
2  M'_{i,j}^t ← sort(M_{i,j}^t);
3  for i ← 1 to length(M'_{i,j}^t) do
4      for j ← 1 to length(M'_{i,j}^t) do
5          temp ← M'[i][j];
6          for i' ← 1 to length(M'_{i,j}^t) do
7              for j' ← 1 to length(M'_{i,j}^t) do
8                  if (M'[i'][j'] == temp) then
9                      M''[i'][j'] ← rank + 1;
10                     rank ← rank + 1;
11                     j' ← length(M'_{i,j}^t);
12                 end
13             end
14         end
15     end
16 end
```

Rank Ascent Identification

The stage of *rank ascent identification* is defined as the strategy used to find the pairs of keywords (w_i, w_j) whose rankings maximise the ascent from timespan t to timespan $(t+1)$.

To pick these pairs of keywords, a matrix $M^{t,t+1}_{rank_{i,j}}$ of size $|v| \times (T-1)$ is created; it stores the difference in ranking of the pairs of keywords between two subsequent timespans t and $(t+1)$. Each entry $\delta^{t,t+1}$ of $M^{t,t+1}_{rank_{i,j}}$ is defined by Equation 6.3 as follows:

$$\delta^{t,t+1} = M_{rank}^{t,t+1}[i][j] = M''[i][j]^t - M''[i][j]^{t+1} \tag{6.3}$$

where $M_{i,j}''^{\,t}$ and $M_{i,j}''^{\,t+1}$ correspond to the ranked matrices returned by Algorithm 1, respectively, for timespans t and $(t+1)$.

If δ is positive, this means that the ranking of the pairs of keywords (w_i, w_j) is ascending (i.e., a *jump* or a *leap*, as will be formally defined in this section). Otherwise, if δ is negative, then it corresponds to a *fall*. This work only focuses on ascents (jumps and leaps) as the aim is to forecast the fast emerging keywords over time. Therefore, the stage of rank ascent identification is reduced to the identification of pairs keywords having ascent in their ranking over time.

Since the ranking ascents have different magnitudes with a minimum of 1, different thresholds for δ are defined in order to study the impact on higher ranking ascents on the identification of emerging keywords. When δ exceeds a certain threshold `, it is defined as a *leap*. Formally, *Leap2Trend* approach defines the different categories of ranking dynamics as following:

$$\delta = \begin{cases} leap, & \text{if } \delta \geq \theta \\ jump, & \text{if } 0 < \delta \leq \theta \\ fall, & otherwise \end{cases} \tag{6.4}$$

Algorithm 2 presents the pseudo-code of the identification of pairs of contextualising keywords, defined as $(w_i, w_j)^*$ that may lead to fast emerging keywords.

Algorithm 2: Rank Ascent Identification

input : Ranked matrices $M_{i,j}''^{\,t}$, $M_{i,j}''^{\,t+1}$, threshold θ
output: contextualising keywords$(w_i, w_j)^*$

```
for t ← 1 to T do
    for i ← 1 to length(M''^t_{i,j}) do
        for j ← 1 to length(M''^t_{i,j}) do
            for i' ← 1 to length(M''^{t+1}_{i,j}) do
                for j' ← 1 to length(M''^{t+1}_{i,j}) do
                    M_rank[i][j] ← M''^t[i][j] − M''^{t+1}[i'][j'];
                end
            end
        end
    end
end
for i ← 1 to length(M_rank) do
    for j ← 1 to T − 1 do
        δ = M_rank[i][j] − M_rank[i + 1][j + 1];
        if (δ > 0 & δ ≤ θ) then
            return (w_i, w_j)*;
        end
    end
end
```

6.2 Experimental Study

This section presents an evaluation of *Leap2Trend* on the task of tracking and detecting contextualising keywords that may lead to emerging research trends. To this end, two datasets related to two research areas: machine learning and bioinformatics were selected. Then, the obtained results of the proposed approach were evaluated on two gold standards: Google Trends hits and Google Scholar citations.

6.2.1 Datasets

The two corpora represent, respectively, 30 years of the NIPS conference papers and 15 years of Medical Image and Computer Assisted Intervention (MICCAI) conference papers.

NIPS Dataset

The NIPS corpora consist of the full text of 7,241 papers published in Neural Information Processing Systems conference between 1987 and 2017. The dataset is described in Section 4.3.1.

MICCAI Dataset

The MICCAI corpora consist of 15 years of Medical Image and Computer Assisted Intervention proceedings from 2004 to 2018 with a total of 3,844 papers. MICCAI is a top conference in the area of bioinformatics. The proceedings were crawled from Springer website[1] under PDF format. Afterward, the text was extracted using the package *"pdftools"*[2] provided by R.

6.2.2 Gold Standards

To evaluate the effectiveness of *Leap2Trend* in forecasting research trends; defined as contextualising keywords, it is needed to find a set of trends determined a priori to be correct; known as gold standard. In the context of this study, this thesis proposes to use both *Google Trends* hits[3] and *Google Scholar* citations[4] as gold standards.

Google Trends has been chosen because it displays search trends data on Google; Google is considered the first place to start for researchers to find background on the research topic[5]. However, Google Scholar has been used to collect the raw citations of publications.

Google Trends Hits

Google Trends analyses the popularity of search queries in Google Search[6] across various regions and languages and it compares the search volume of different queries over time[7].

[1] https://www.springer.com/
[2] https://cran.r-project.org/web/packages/pdftools/pdftools.pdf
[3] https://trends.google.com/
[4] https://scholar.google.com
[5] library.royalroads.ca/infoquest-tutorials/internet-searching/google-vs-google-scholar-which-one-do-i-use
[6] https://www.google.com/
[7] https://en.wikipedia.org/wiki/Google_Trends

Due to its ability to track various words and phrases that are typed into Google's search-box over time, it has been found that Google Trends aligns with *Leap2Trend* that tracks the co-occurrence of pairs of scientific keywords over time, which may lead to emerging trends. To this end, the following methodology is proposed to compare the results of *Leap2Trend* with Google Trends hits:

1. For each pair of keywords studied by *Leap2Trend*, the results from Google Trends were downloaded. These results report the Google query volumes of this pair of keywords. Recall that the keywords are typed as they are in the interface of Google Trends without quotations for more than one-word keywords. The API *pytrends*[8] was used. This API downloads data in form of csv files recording the number of queries of this pair of keywords on a monthly basis. For convenience, the number of queries of Google Trends will be referred as *Google Trends hits*. The parameter *'timeframe'* of *pytrends* is set to (2004-2017) and (2004-2018) for the NIPS and MICCAI corpora, respectively, respecting the time-frame of both corpora as described in Section 6.2.1. The start date 2004 is justified by the start of Google Trends service. For this reason, the NIPS set of publications before 2004 were ignored when the proposed approach has been evaluated. For the parameter *'geo'* that refers to the region of search, It was set to the by-default parameter, which returns worldwide results. Recall that the retrieval time of Google Trends data was November and December 2018.

2. To be consistent to the results provided by *Leap2Trend* on yearly basis, the Google Trends hits have been aggregated in the csv files by summing up the hits of each 12 months together.

3. Referring to Section 6.1.4, the ascents (*jumps* and *leaps*) in ranking over time of each pair of keywords were defined. For each ascent, the Google Trends hits have been tracked 3 years ahead and the slope of the linear regression of these hits has been computed. The aim behind this computation is to check if the jump in ranking captured by *Leap2Trend* indicates a positive slope and consequently defines this pair of keywords as contextualising keywords. This could show the predictive power of the proposed approach in forecasting trends. The choice of 3 years as a duration is justified in Section 6.1.1 and the slope m_{hits} of the linear regression of Google Trends hits is defined as follows:

$$m_{hits} = \frac{\sum_{i=1}^{4}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{4}(x_i - \bar{x})^2} \tag{6.5}$$

   where x and y correspond, respectively, to the year of hits and the number of hits, $\bar{x}$ and $\bar{y}$ represent, respectively, their means. The number 4 corresponds to the number of years to consider starting from the year of the ascent and 3 years ahead.

Google Scholar Citations

Google Scholar, in January 2018, was considered the world's largest academic search engine, with roughly 389 million documents indexed including articles, citations and patents (Gusenbauer, 2018).

[8] https://github.com/GeneralMills/pytrends

Due to its ability to calculate and display the citation counts of scientific publications and its wide coverage of article published in English with an estimate of 100 million[9], Google Scholar is used to extract the raw citations of the NIPS and MICCAI publications used in this evaluation. To do so, the software *Public or Perish*[10] is used, which uses Google Scholar to obtain the raw citations.

The evaluation methodology of *Leap2Trend* against this gold standard has two steps:

1. For each pair of keywords studied by *Leap2Trend*, the set of all publications mentioning these keywords in their titles were selected and the total number of their citation counts returned by Google Scholar was computed; assuming that the title plays a pivotal role in communicating research.

2. The ascents of these pairs of keywords with the citation counts over timespans were computed. A good result of *Leap2Trend* corresponds to a positive correlation between the ascents and the citations, i.e., when the jump increases, the citation count increases and vice versa.

6.2.3 Evaluation Metrics

The performance of *Leap2Trend* results is assessed against the two gold standards defined above by means of *ascent accuracy*, *ascent recall* and *ascent precision*. The goal in this evaluation is to answer the following two questions:

i How accurate is *Leap2Trend* in detecting contextualising keywords, and accordingly in predicting future trends at an early stage?

ii How precise is *Leap2Trend* in following the flow of Google Trends hits and citation counts?

Ascent Accuracy and Recall

Ascent Accuracy. The ascent accuracy (*accuracy*) evaluates the prediction power of *Leap2Trend* in detecting contextualising keywords at an early stage by tracking the ascents in ranking of pairs of keywords that will eventually lead to emerging trends. Therefore, the accuracy is defined as the fraction of the number of $ascents^+$ – defined as the ascents that successfully led to positive slopes in the linear regression of Google Trends hits – with the number of ascents returned by *Leap2Trend* as expressed in Equation 6.6.

$$accuracy = \frac{|\{ascents^+\} \cap \{ascents\}|}{|\{ascents\}|} \tag{6.6}$$

Ascent Recall. The ascent recall (*recall*) attests the number of ascents in the gold standard that were successfully detected by *Leap2Trend*. Therefore, the recall is defined as the fraction of the number of *ascents* returned by *Leap2Trend* with the number of $ascents^+$ that successfully led to positive slopes in the linear regression of Google Trends hits, as expressed in Equation 6.7.

$$recall = \frac{|\{ascents^+\} \cap \{ascents\}|}{|\{ascents^+\}|} \tag{6.7}$$

[9] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0093949
[10] https://harzing.com/resources/publish-or-perish

Ascent Precision

Two measures of ascent precision are defined to evaluate the results of *Leap2Trend*. The first measure P_{GT} evaluates the obtained results against Google Trends hits, while the second measure P_{GS} evaluates the results against Google Scholar citations. Both of them refer to how close are *Leap2Trend* ascents to Google Trends hits or citation counts. Close means how the ascents are positively correlated with the hits or the citations. Therefore, each of these two precision measures is formally defined as the fraction of the number of $(w_i, w_j)^{corr+}$ – the pairs of keywords having positive correlations with the hits or citations – with the total number of pairs of keywords formed from the vocabulary v, $v \subseteq \mathscr{V}$ as defined in Section 6.1.3. Equation 6.8 expresses the ascent precision (*precision*), where *precision* refers to P_{GT} or P_{GS} with respect to the used gold standard.

$$precision = \frac{|\{(w_i, w_j)^{corr+}\}|}{|\{(w_i, w_j)\}|}; i, j \in |v| \tag{6.8}$$

To measure the correlation between the ascents and the hits or the citations, the *Spearman's correlation coefficient* ρ is used. ρ computes the strength and the direction of association between the ascents and any of the hits or citations as follows:

$$\rho = \frac{\sum_s (x_s - \bar{x})(y_s - \bar{y})}{\sqrt{\sum_s (x_s - \bar{x})^2 \sum_s (y_s - \bar{y})^2}} \tag{6.9}$$

where s is the paired score $(ascent, GoogleTrendHit)$ or $(ascent, citation\_counts)$, x corresponds to the hits or to the citation counts and y corresponds to the ascents, $\bar{x}$ corresponds, respectively, to the mean of hits or the mean of citations counts and $\bar{y}$ corresponds to the mean of ascents.

6.2.4 Results

For each of the used datasets, namely the NIPS and MICCAI, three series of experiments were ran within *Leap2Trend* approach following the three temporal embedding paradigms described in Section 6.1.2. Then, the obtained results were evaluated against the two gold standards: Google Trends hits and Google Scholar citations defined in Section 6.2.2.

For both datasets, the selection of the keywords of interest to be studied in this research is firstly done. To do so, the first step consists the selection of the top 100 frequent bigrams extracted from the titles of the publications. Bigrams were used rather than unigrams because of their frequent use in scientific corpora especially in machine learning and bioinformatics; the two research areas this chapter studies. The selection of these keywords from the titles is justified by the fact that the title of a scientific paper is mostly self-explanatory reflecting the work being reported; hence it possibly contains the important keywords of interest in any research area. From these 100 bigrams, only the bigrams whose combination provides available information from Google Trends were kept in order to fairly evaluate *Leap2Trend* against the gold standard. The combination of pairs of bigrams does not necessarily mean that the two bigrams appear together in the same paper title or the same Google query. In this thesis, the combination of these pairs of bigrams is based on their frequencies, and the availability of related information from Google Trends. This restricted the keywords of interest to only 20 bigrams. This number has been also supported by

Google Hot Trends[11] that displays the 20 hot and fastest rising search terms at a time. Similarly, the aim is to early detect the fastest rising trends in the field of study. The number of these emerging trends could not be high as the tracking of the evolution is on yearly basis. For instance, a study performed by Hoonlor *et al.* (Hoonlor et al., 2013) on evolving computer science research showed that the average length of the evolutionary chain is 4.5 years with few new topics. This has been also proved by a study conducted by Asooja *et al.* (Asooja et al., 2016) on the domain of Natural Language Processing, Information Retrieval, and Semantic Web. They detected only two new topics in a period of 6 years from 2008 to 2014. Recall that the same set of pairs of keywords is used for all timespans in order to keep tracking their similarities/dissimilarities over time. It is very likely that this approach prevents to include new keywords/topics that may appear when time progresses. But, the main goal is to provide a prove of concept for the proposed approach *Leap2Trend* and proves its ability to early detect emerging trends.

After preprocessing both the NIPS and MICCAI corpora, the skip-gram model is trained at every timespan with the embedding dimension $N = 200$ and the context window $= 6$. The choice of these hyperparameters is supported by the previous findings (Dridi et al., 2018) detailed in Chapter 4 that showed that these hyperparameters are optimal within scientific corpora. Recall that *word2vec package* of the open source *Gensim Python Library*[12] has been used to implement the word vector representations. *Gensim* was ran on Windows Intel core i7 platform that supports *Python* and *NumPy*. For the incremental windows, two trainings were performed. The first training follows an updated embedding as described in Section 6.1.2 while the second training created a fresh trained model by re-training it from scratch. The code of these two trainings is publicly available here[13]. For the sliding windows, the model was trained at every timespan, because the sliding paradigm results in new vocabulary forgetting one year vocabulary and adding one year ahead vocabulary as shown in Figure 6.3.

After each training at a timespan t, a similarity matrix $M^t_{i,j}$ is created as described in Section 6.1.3 that corresponds to the 20 keywords of interest extracted from the titles of the publications as described above. At every timespan t, the similarity values of $M^t_{i,j}$ were ranked and then $M^{t,t+1}_{rank_{i,j}}$ was created; it stores the difference in ranking of the pairs of keywords between two subsequent timespans t and $(t+1)$. For each pair of keywords, all ascents were picked; those corresponding to a positive δ calculated following Equation 6.3.

Leap2Trend vs Google Trends Hits

For each ascent, the slope of the linear regression of Google Trends hits is computed as expressed in Equation 6.5. In order to avoid bias, the ascent picked at 2005 were ignored, because it corresponds to the ascent in ranking of the pair of keywords between 2004 and 2005 while δ at 2004 is set to 0 (2004 is the starting year of analysis and corresponds to the starting year of Google Trends). After the selection of all ascents related to all studied pairs of keywords, the related accuracy is computed as described in Equation 6.6. This accuracy corresponds to any ascent. Then, different thresholds for δ were set: $\{5, 10, 20, 30\}$ defining leaps with various magnitudes.

[11] https://en.wikipedia.org/wiki/Google_Trends
[12] https://radimrehurek.com/gensim/models/word2vec.html
[13] https://github.com/AmnaKRDB/Leap2Trend

The choice of these thresholds was based on the overall obtained values of δ on both datasets after the three training paradigms. For this reason, some of these thresholds may not be found on some results such as the thresholds 20 and 30 in the *fresh embedding* of the MICCAI dataset as shown in Figure 6.6.

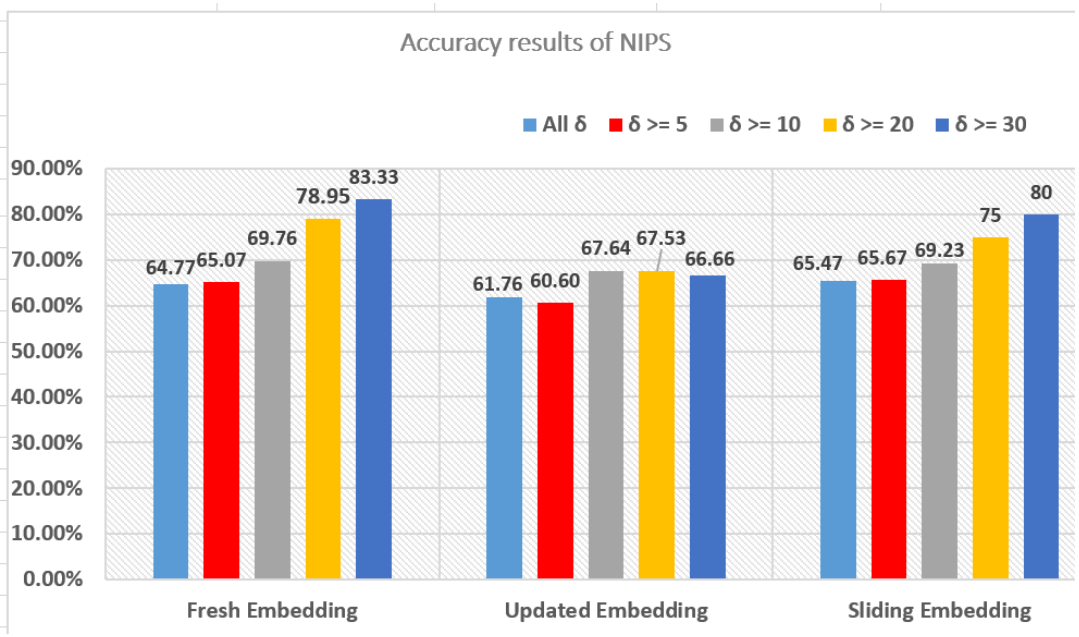


FIGURE 6.5: Accuracy results of Leap2Trend based on the NIPS with respect to the three embedding paradigms and different thresholds of $\delta, \delta > 0$ in all cases

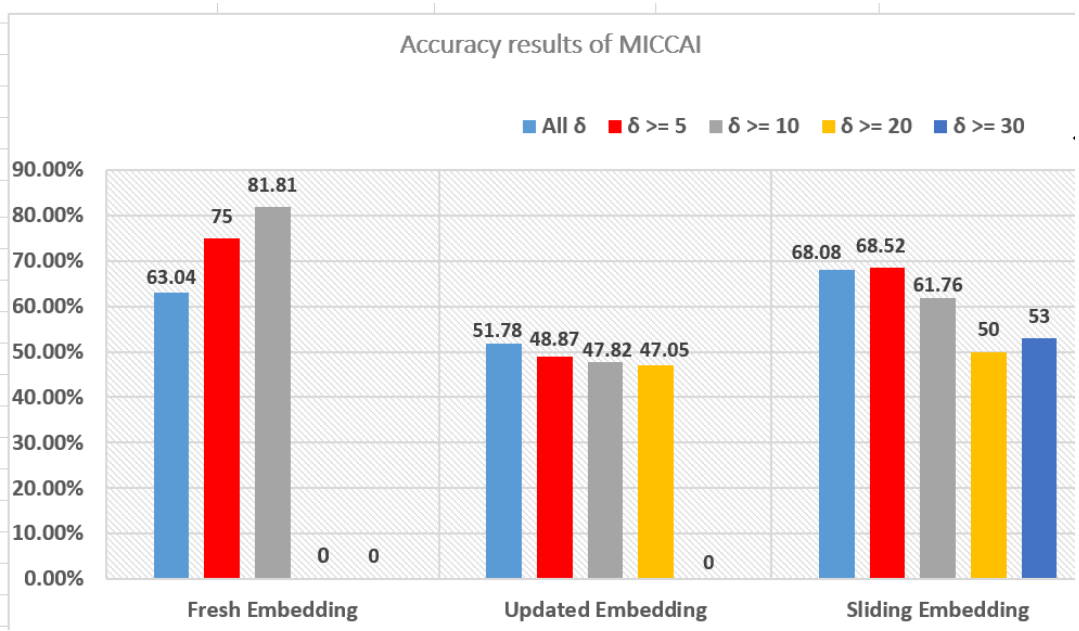


FIGURE 6.6: Accuracy results of Leap2Trend based on the MICCAI with respect to the three embedding paradigms and different thresholds of $\delta, \delta > 0$ in all cases

Figure 6.5 and Figure 6.6 show accuracy measures of *Leap2Trend* with the three

embedding paradigms: *fresh embedding*, *updated embedding* and *sliding embedding*, and with different thresholds of δ applied to the NIPS and MICCAI datasets. According to these accuracy results computed based of Google Trends hits as gold standard, *Leap2Trend* shows promising findings in forecasting research trends in different domains. For instance, the accuracy is above 63% in all different settings of the fresh embedding and it exceeds 80% in some cases.

The overall results shown in Figure 6.5 and Figure 6.6 reveal that the best accuracy is given when experimenting (i) with the fresh embedding and (ii) with high leaps. (i) could be justified by the fact that the ideal approach for incremental embedding would be to retrain the model from scratch including new vocabulary in the training corpus (Kaji and Kobayashi, 2017). That is because the incremental training of word embeddings may drift words learned from later batches arbitrary far from words in earlier batches that are not re-presented. This observation is supported by the obtained results on both the NIPS and MICCAI datasets, where the updated embedding performed the worst in all settings. (ii) highlights the importance of the magnitude of ascents; when the ascent increases, the accuracy increases accordingly. The substantial improvement in accuracy from any δ to a greater threshold underlines the ability of *Leap2Trend* to accurately detect contextualising keywords at a very early stage by paying attention to the ascents in ranking of pair of keywords over time.

To validate the observation on the importance of the magnitude of the ascents, the average of slopes Δm_{hits} is computed at every threshold δ. The average slope Δm_{hits} corresponds to the fraction of the sum of the slopes m_{hits} with the number of detected ascents $|\{ascents\}|$ and it is expressed by Equation 6.10 as follows:

$$\Delta m_{hits} = \frac{\sum_{i=1}^{|\{ascents\}|} m_{hits}}{|\{ascents\}|} \tag{6.10}$$

Figure 6.7 and Figure 6.8 illustrate the obtained results of average slopes on the NIPS and MICCAI datasets, respectively. Similar to the previous results of accuracy, the fresh embedding performs the best in both datasets. For instance, the average of slopes Δm_{hits} gradually increases with the increase of ascents. However, for the sliding embedding related to the MICCAI dataset, a decrease in Δm_{hits} is noticed starting from the threshold $\delta \geq 10$. This is justified by the rarity of picked ascents with higher magnitude. As a matter of fact, this decrease goes in parallel with the accuracy that drops to 50% with $\delta \geq 20$ as shown in Figure 6.6. In reality, this 50% represents 4 positive slopes over 8 detected ascents with more than 20 ascents. Therefore, both the average of slopes and the accuracy are highly sensitive to the magnitude of ascents.

For the updated embedding, the obtained average of slopes is the worst. This supports the previous obtained results on accuracy and confirms the assumption that the ideal approach for incremental embedding would be to retrain the model from scratch. But, it is worth mentioning that the updated embedding is more efficient than the fresh embedding. This is obvious as retraining the model comes at cost in time.

For overall experimental results on the NIPS and MICCAI datasets, *Leap2Trend* shows a great potential to early detect contextualising keywords leading to emerging research trends, both quantitatively (accuracy) and qualitatively (average slope). *Leap2Trend* achieves this by tracking ascents and setting different thresholds that are used as indicators to detect the contextualising keywords.

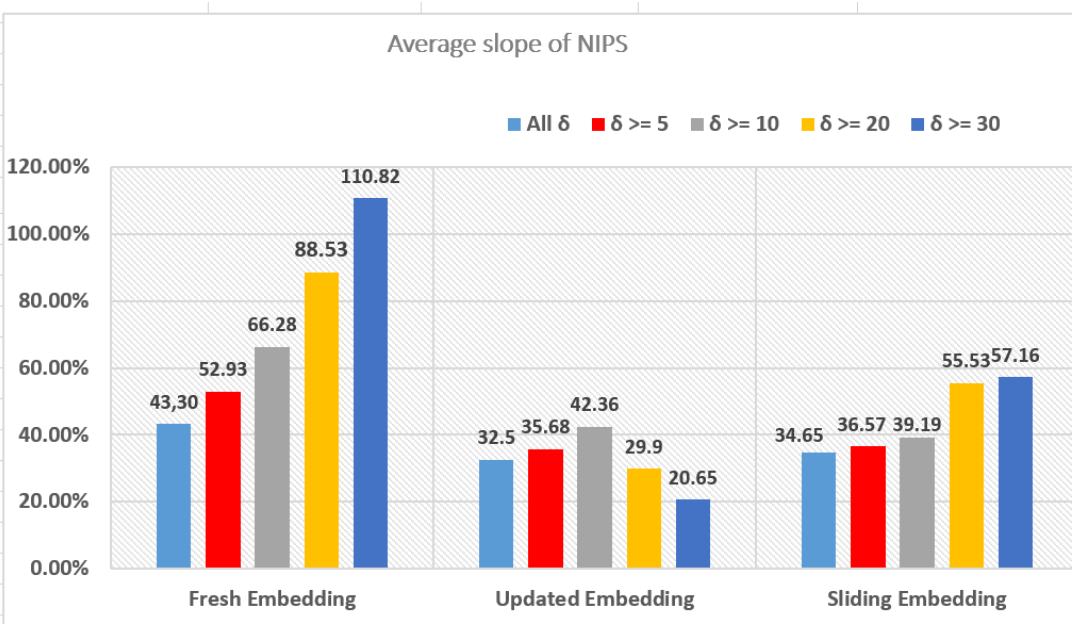


FIGURE 6.7: Average slope results of Leap2Trend based on the NIPS dataset with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

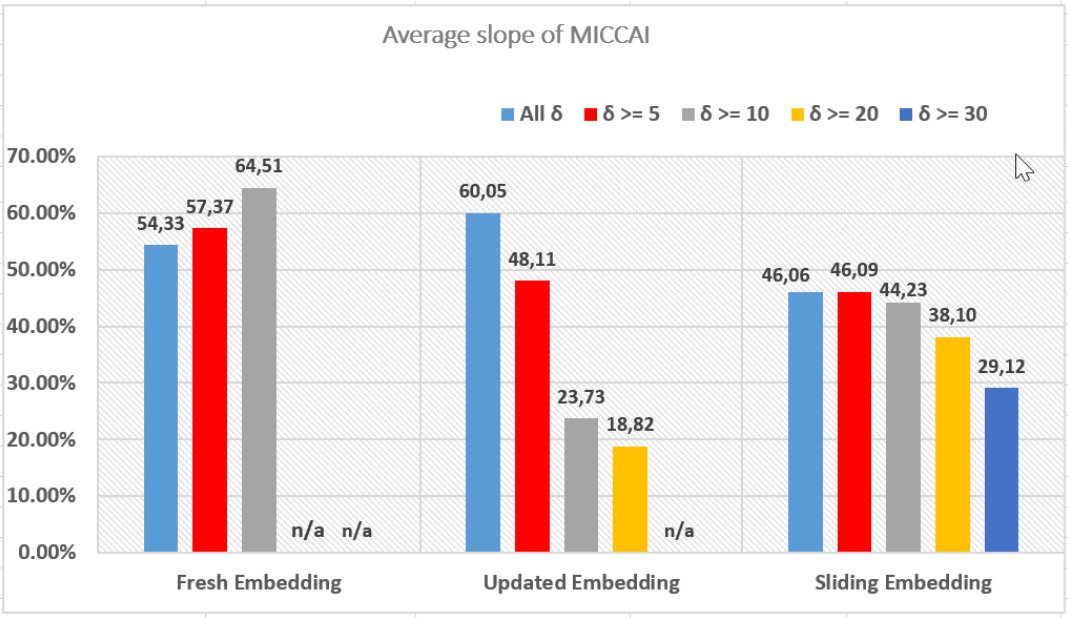


FIGURE 6.8: Average slope results of Leap2Trend based on the MICCAI with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

As a proof of evidence, *Leap2Trend*, applied to the NIPS dataset with the fresh embedding, detects an ascent of $\delta = 11$ of the pair of keywords (*"neural_network – machine_learning"*) between 2012 and 2013 as shown in Figure 6.10(a). This ascent is highly significant as this pair of keywords reflects that *machine learning* is highly relying on *neural networks* during this time period. This is insightful as it shows that *Leap2Trend* is able to detect the frequent co-occurrence of these pairs of keywords in

the same context, which may lead to the hybridisation of these two keywords. In fact, the average of slope Δm_{hits}, which is equal to 76.9 supports this assumption. The strength of this slope is clearly shown in Figure 6.10(a) as the number of Google Trends hits passes from 86 in the year when the ascent happened to 329 after 3 years. Similarly, *Leap2Trend*, applied to the MICCAI with the fresh embedding, detects an ascent of $\delta = 10$ of the pair of keywords (*"lung_cancer – breast_cancer"*) between 2009 and 2010. This ascent was insightful as the statistics on medical research in 2010 showed that *lung cancer* was the most second commonly diagnosed cancer in the UK after *breast cancer*[14]. This could justify why *Leap2Trend* detected the ascent of these two keywords as they co-occur together. This observation is then supported by the average of slope Δm_{hits}, which is equal to 40.1 and shows an increase in Google Trends hits in Figure 6.8.

Overall, the accuracy results on the NIPS and MICCAI datasets show a great potential of the proposed approach *Leap2Trend* to detect research trends early. It is also important to reveal how many of the ascents presented in the gold data were detected by *Leap2Trend*. To do so, the recall is computed as defined in Equation 6.7. A relevant ascent, named as $ascent^{+}$, is defined as an ascent approved by a positive slope of the Google Trends hits looking three years ahead. Figure 6.9 shows recall measures of *Leap2Trend* with the three embedding paradigms applied to the NIPS and MICCAI datasets. The overall results show promising findings in early recalling research trends. For instance, the recall is above 50% in all settings on the NIPS dataset, and it reaches and exceeds 40% on the MICCAI dataset. *Leap2Trend* reveals then a great potential to recall trends ahead in time. The obtained recall results on both datasets align with the accuracy results on the impact of every embedding setting. As a matter of fact, the fresh embedding performs the best with the NIPS dataset with 57.79% while the sliding embedding performs the best with the MICCAI dataset with 43.83% for the same reasons detailed for the accuracy. Exceptionally for recall with the MICCAI, the two embedding settings (fresh and updated) perform similarly with 39.72%. This could be justified by the size of corpora as the MICCAI has small corpora with more likely few new keywords, which makes the incremental embedding less sensitive to the followed paradigm whether it is fresh or updated.

After testing the effectiveness of *Leap2Trend* in early predicting research trends using accuracy, the closeness of *Leap2Trend* ascents to Google Trends hits is tested by performing a fine-grained analysis. This fine-grained analysis aims to check to what extent the ascents of *Leap2Trend* are correlated with Google Trends hits. To do so, the Spearman's correlation coefficient (Equation 6.9) is computed to every pair of keywords. Afterward, the precision P_{GT} is measured following Equation 6.8. Figure 6.11 illustrates the obtained precision results on the NIPS and MICCAI datasets with the three embedding paradigms.

Interestingly, these results indicate that the sliding embedding in both datasets performs significantly better than the incremental embedding (the fresh embedding and the updated embedding) with a precision of 88.88% and 61.53% for the NIPS and MICCAI datasets, respectively. This could be justified by the fact that the sliding window of 3 years length could perfectly match the keywords published in the papers with the keywords used in Google Search unlike the incremental window that keeps the old vocabulary. This affects the similarity of keywords and consequently affects their ranking and hence their ascents. For the updated embedding,

[14] https://www.bci.qmul.ac.uk/en/our-research/lung-cancer

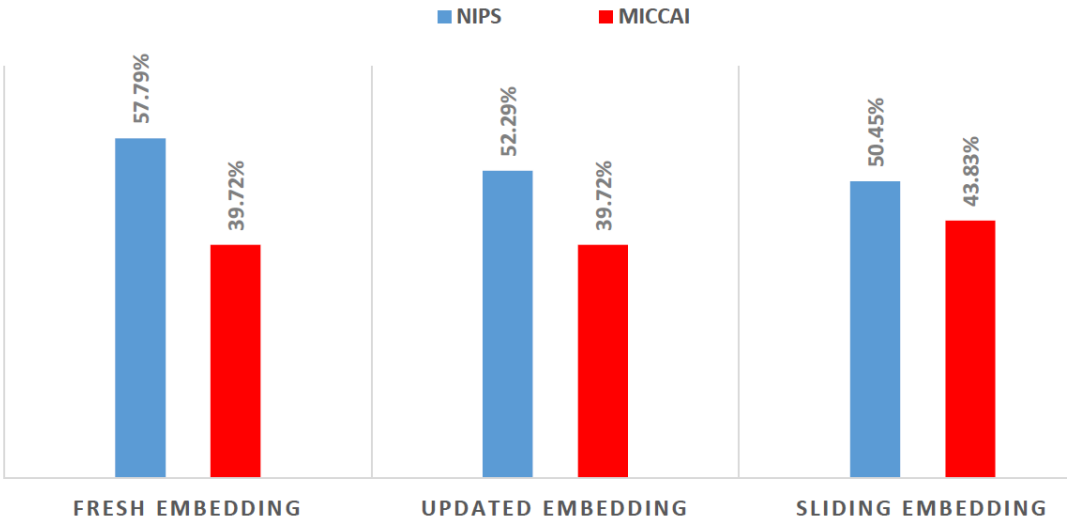
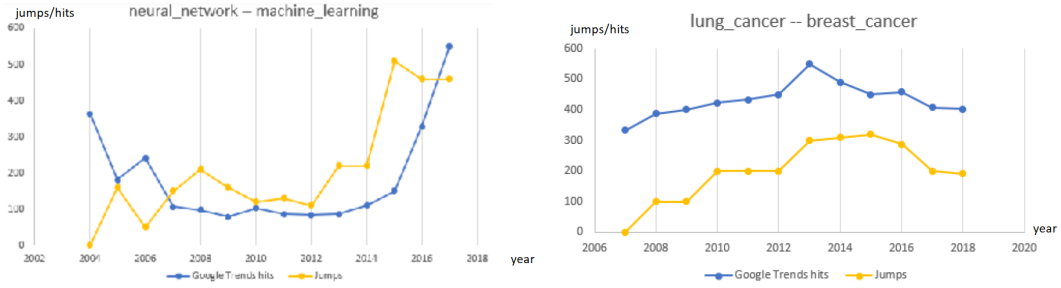


FIGURE 6.9: Recall results of Leap2Trend based on the NIPS and MICCAI datasets against Google Trends hits



(a) NIPS Example (b) MICCAI Example

FIGURE 6.10: The linear regression of jumps and Google Trends hits related to two pairs of keywords from the NIPS and MICCAI datasets[a=]

[a=]For visualisation purpose, the values of jumps were multiplied by 10 in order to clearly display the jumps with respect to the Google Trends hits

the precision results confirm those previously obtained with accuracy; it performs worst with all measures of effectiveness. Overall, the P_{GT} results support the accuracy ones and show that the proposed approach *Leap2Trend* is able to early forecast trends matching Google Trends hits. For instance, the Spearman's correlation coefficient shows a strong correlation between *Leap2Trend* ascents and Google Trends hits for the sliding embedding with 65% and 55% of ρ values greater than 0.6 for, respectively, the NIPS and MICCAI datasets.

For all the settings and measures, *Leap2Trend* performs better on the NIPS dataset than the MICCAI dataset. This could be justified by two reasons. The first reason refers to the size of corpora; the NIPS corpora is much bigger than the MICCAI corpora and it has been proved in the literature (Mikolov et al., 2013d; Mikolov et al., 2013b) that word embedding quality increases as the corpus size increases. The second reason may refer to the popularity and the strength of the conference. For instance, the NIPS conference is more than 30 years old while the MICCAI is only 21 years old. The prestige of the conference contributes to its strength and rapidity in

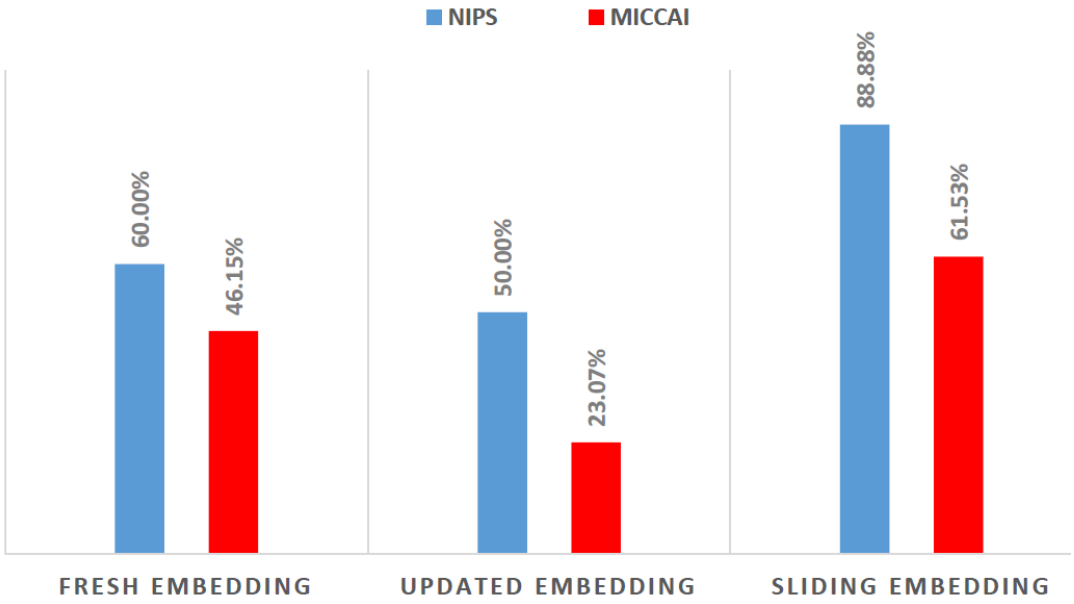


FIGURE 6.11: Precision of Leap2Trend based on the NIPS and MICCAI datasets against Google Trends hits

developing new research topics.

Leap2Trend vs Google Scholar Citations

In order to support *Leap2Trend* findings obtained against the gold standard Google Trends hits, a new validation of *Leap2Trend* results is performed with the citation counting approach, which is widely used in the literature and provides a snapshot of a fast-growing field. To do so, academic citations from Google Scholar were retrieved for all the NIPS and MICCAI publications as described in Section 6.2.2. Then, the ascents of all studied pairs of keywords, from NIPS and MICCAI datasets over the three embedding paradigms, were compared with citation counts.

Similar to the obtained results performed with Google Trends hits, Spearman's correlation coefficient (Equation 6.9) was used to measure the correlation between the ascents and the citation counts. Afterward, the precision P_{GS} of *Leap2Trend* results on the NIPS and MICCAI datasets was computed following Equation 6.8.

Figure 6.12 demonstrates the obtained precision results on the NIPS and MICCAI datasets with the three embedding paradigms. According to these results, the two incremental embeddings outperform the sliding embedding in both datasets with a precision that reaches 90% with the fresh embedding applied to the NIPS dataset. These results are meaningful because the incremental embedding keeps the history of publications, which affects the similarity of keywords and consequently affects their ascents. This perfectly matches the citation counting approach that takes time to progress and reveal trends. However, the sliding embedding refers to only 3 years publications with a forgotten one year publications and an added new one year publications. This window size is not enough to reflect the citation counts that need time to evolve.

Overall, *Leap2Trend* precision results against Google Scholar citations support the previous results on Google Trends hits as well as accuracy, and they show the

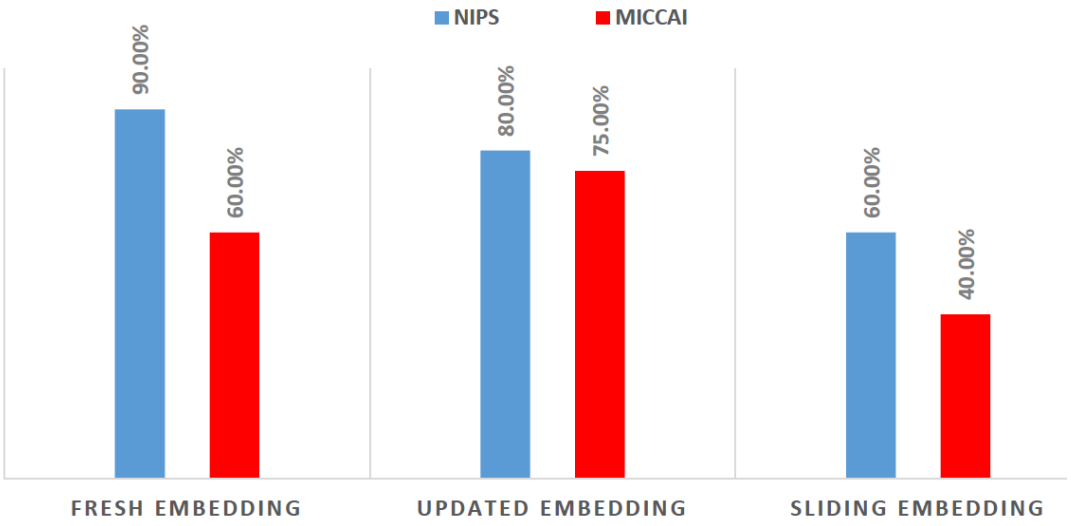


FIGURE 6.12: Precision of Leap2Trend based on the NIPS and MICCAI datasets against Google Scholar citations

effectiveness of the proposed approach *Leap2Trend* to detect emerging trends with promising findings.

6.3 Summary

In this chapter, the computational history of science has been performed through the detection of *contextualising keywords* that may lead to new scientific trends. For this end, *Leap2Trend* has been proposed. It is a new approach for early detection of research trends. *Leap2Trend* has harnessed word embedding techniques to dig into the paper content and track the dynamics of similarities between pairs of keywords. To do so, *Leap2Trend* trained temporal embeddings following two temporal paradigms: incremental and sliding. Then, after each training, it created a similarity matrix that stores the similarities of pairs of keywords of interest. Afterward, it ranked the entries of this matrix and computed the ascents in ranking over different timespans. Finally, for each picked ascent, *Leap2Trend* performed different evaluations against *Google Trends hits* and *Google Scholar citations* in order to test if the detected ascent of the pair of keywords refers to a new emerging trend. The obtained results showed the effectiveness of *Leap2Trend* to early detect emerging keywords.

The major contributions of this chapter are listed as follows:

1. Introducing *Leap2Trend*, a new framework for the detection of new research trends at a very early stage by tracking the *contextualising keywords*.
2. Leveraging temporal word embedding techniques, namely word2vec (Mikolov et al., 2013d) for fine-grained content analysis of scientific corpora.
3. Applying *Leap2Trend* to real-world datasets in two research areas: machine learning and bioinformatics, which could give insights about the validity and the generalisability of the proposed approach.
4. Validating the approach using Google Trends hits and Google Scholar citations as gold standards.

This chapter has followed the first path to perform the computational history of science, which is devoted to the detection of new scientific trends. However, the new scientific trends in this chapter are defined as the *contextualising keywords* that start to frequently co-occur together in the same context. To detect these *contextualising keywords*, the dynamics of similarities between pairs of keywords have been tracked by adopting the rankings of similarities and computing the ascents in ranking over different timespans. In the next chapter, the second path to perform the computational history of science will be followed, which is devoted to tracking the dynamism of scientific keywords by studying the evolvement of their semantic neighbourhood over time.

Chapter 7

Vec2Dynamics: Tracking The Dynamism of Scientific Keywords

> "The test of our progress is not whether I add more to the abundance of those who have much, it is whether I provide enough for those who have little."
>
> — Franklin D. Roosevelt. (1882–1945)

In the previous two chapters, the computational history of science has been performed by tracking the emerging scientific trends, which were defined as *converging keywords* and *contextualising keywords*. This was done by computing the similarities between selected pairs of keywords and tracking the changing in similarity over time. In this chapter, the computational history of science goes beyond the tracking of emerging scientific keywords. But, it concerns tracking the evolvement of their semantic neighbourhood over time, which gives a more generic view on the dynamics of science including *emerging*, *dying*, *recurrent* and *persistent* keywords. This chapter introduces *Vec2Dynamics*, a temporal word embedding approach that reports the *stability of k-nearest neighbors (k-NN)* of scientific keywords over time; the stability indicates whether the keywords are taking new neighborhood due to evolution of scientific literature. To evaluate how *Vec2Dynamics* models such relationships in the domain of *machine learning*, scientific corpora from the papers published in the Neural Information Processing Systems (NIPS) conference between 1987 and 2016 have been used. The descriptive analysis performed in this chapter verifies the efficacy of the proposed approach. In fact, a generally good consistency between the obtained results and the *machine learning timeline*[1] was detected.

The rest of this chapter is organised as follows. Section 7.1 details the *Vec2Dynamics* approach and its different stages. Section 7.2 describes the evaluation of the proposed approach, presents and discusses the obtained results. Finally, Section 7.3 summarises the chapter. The work described in this chapter is submitted to *machine learning journal*.

7.1 Vec2Dynamics

In order to understand and uncover the dynamics of scientific literature, *Vec2Dynamics* is proposed – a fine-grained content analysis approach that relies on

[1] https://en.wikipedia.org/wiki/Timeline_of_machine_learning

temporal word embeddings to delve into the content of research papers.

First, *Vec2Dynamics* digs into the textual content by applying word2vec embeddings (Mikolov et al., 2013d). Then, it grasps dynamic change in interest and popularity of research topics by iteratively applying *k-NN stability* to the keywords/topics of interest over time and accordingly capturing the *recurrent*, *non-recurrent*, *persistent* and *emerging* keywords. Formal definitions of these types of keywords are given later in this section. The general architecture of *Vec2Dynamics* is first described and then the functionalities of its different stages are detailed.

7.1.1 Vec2Dynamics Architecture

The architecture of *Vec2Dynamics* is depicted in Figure 7.1. The whole model can be divided into four different stages.

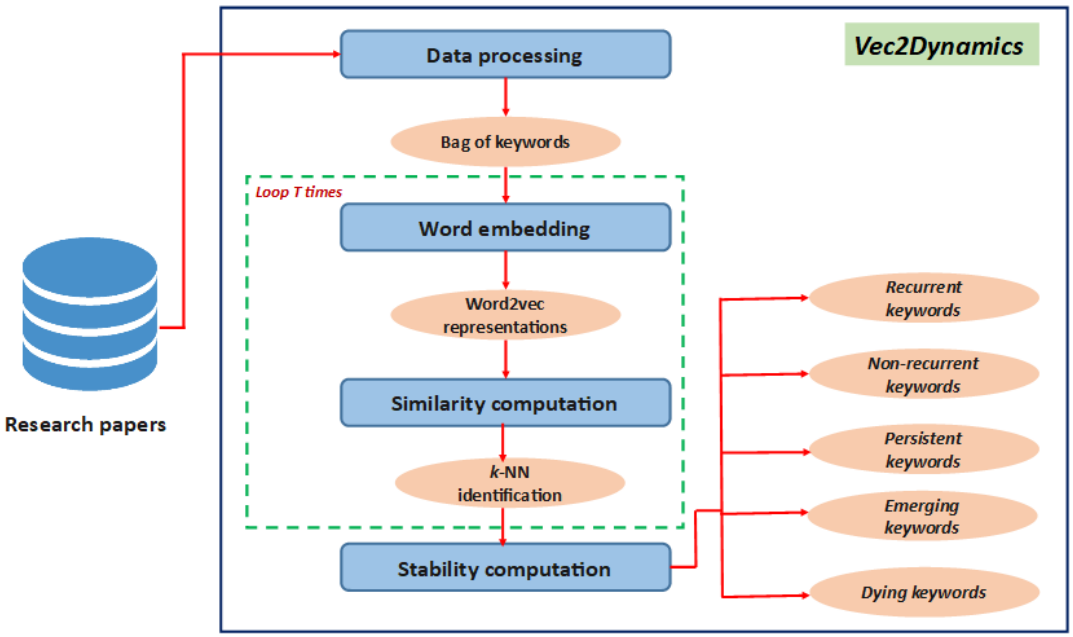


FIGURE 7.1: Workflow of Vec2Dynamics

i **Data preprocessing.** At this stage, the textual content of research papers is preprocessed and cleaned up taking into account the specificity of scientific language. For instance, the frequent use of bigrams in scientific language is considered such as *"information system"*, and *"artificial intelligence"*, and a bag of keywords is constructed, where keywords are either *unigrams* or *bigrams*. Data preprocessing consists then of two steps: (a) the removal of stop words; and (b) the construction of bag of words, where words are either unigrams or bigrams. More details on the data preprocessing stage are given in Section 4.3.1.

ii **Word embedding.** At this stage, the skip-gram architecture of word2vec (Mikolov et al., 2013d) is adopted to learn word vectors over time. This stage is repeated for each corpus $P_{t,t=1,...,T}$ that corresponds to the corpus of all research papers in the t^{th} timespan. More details will be given in Section 7.1.2.

iii **Similarity computation.** After generating the vector representation of keywords, *cosine similarity* between embedding vectors is applied to find the k-nearest neighbors of each keyword. Recall that cosine similarity between two keywords w_i and w_j refers to the cosine measure between embedding vectors u_{w_i} and u_{w_j} as follows:

$$similarity(w_i, w_j) = cosine(u_{w_i}, u_{w_j}) = \frac{u_{w_i} \cdot u_{w_j}}{\|u_{w_i}\| \cdot \|u_{w_j}\|} \tag{7.1}$$

As with the previous stage, this stage of similarity computation is also repeated at each timespan t.

iv **Stability computation.** At this stage, the stability of k-NN of each keyword of interest is studied over time in order to track the dynamics of the scientific literature. To do so, a *stability measure* is defined (Equation 7.2), which could be computed between the sets of k-NN keywords over two subsequent timespans t and $(t+1)$. Based on the obtained stability values, four types of keywords/topics are defined: *recurrent*, *non recurrent*, *persistent* and *emerging* keywords. More details will be given in Section 7.1.3.

7.1.2 Temporal Word Embeddings

Vec2Dynamics relies on a central stage of word embeddings that learns word vectors in a temporal fashion in order to track the dynamics of scientific literature over time. To this end, the skip-gram architecture of word2vec (Mikolov et al., 2013d) is adopted, which aims to predict the *context* given a word w_i. Note that the *context* is the span of words within a certain range before and after the current word w_i (Mikolov et al., 2013e).

Notation

Let consider corpora of research papers collected across time. Formally, let $P = (P_1, P_2, \ldots, P_T)$ represents the used corpora, where each P_t is the corpus of all papers in the t^{th} timespan, and $\mathcal{V} = (w_1, w_2 \ldots, w_V)$ the vocabulary that consists of V words present in P. It is possible that some $w_i \in \mathcal{V}$ do not appear at all in some P_t. This happens because new keywords emerge while some old keywords die; something that is typical of scientific corpora. Let V_t denote the vocabulary that corresponds to P_t and $|V_t|$ denote the corresponding vocabulary size used in training word embeddings at the t^{th} timespan.

Given this time-tagged scientific corpora, the goal is to find a dense, low-dimensional vector representation $u^t_{w_i} \in \mathbb{R}^N$, $N \ll V_t$ for each word $w_i \in V_t$ at each timespan $t = \{1, \ldots, T\}$. N is the *dimensionality* of word vectors that corresponds to the length of the vector representations of words.

Skip-gram Model

The architecture of the skip-gram model is detailed in Chapter 3, Section 3.3.1. To tune the hyperparameters of skip-gram model, the approach of *k-NN embedding stability* detailed in Chapter 4 is followed. This approach showed that the optimal hyperparameters are, respectively, 200 and 6 for vector dimensionality N and the context window for scientific corpora.

7.1.3 k-NN Stability

After learning temporal word embeddings on the scientific corpora, *Vec2Dynamics* uses the *stability of k-nearest neighbors (k-NN)* of word vectors as the objective to measure while tracking the dynamics of research keywords/topics. The k-NN keywords of a target keyword w_i correspond to the k keywords similar to w_i. Recall that cosine similarity is used to calculate the similarity between two keywords w_i and w_j.

Notation

Let $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$ denote, respectively, the sets of k-NN of the keyword w_i over two successive timespans $(t-1)$ and t. $\Psi^{t}_{w_i}$ denotes the k-NN stability of w_i at the timespan t as the logarithmic ratio of (a) the intersection between the two sets $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$ to (b) the difference between them. Formally, $\Psi^{t}_{w_i}$ is defined as follows:

$$\Psi^{t}_{w_i} = \log_k \left(\frac{|S^{t}_{w_i} \cap S^{t-1}_{w_i}|}{0.5 \times |S^{t}_{w_i} \ominus S^{t-1}_{w_i}|} \right) \tag{7.2}$$

Recall that the two sets $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$ are equal and their differences are symmetric as $|S^{t-1}_{w_i}| = |S^{t}_{w_i}| = k$, which justifies the multiplication by 0.5 in the denominator.

The k-NN stability $\Psi^{t}_{w_i}$ ranges from -1 to $+1$. -1 (corresponding to $\log_k(\frac{1}{k})$) refers to the case, where no intersection exists between $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$ ($|S^{t}_{w_i} \cap S^{t-1}_{w_i}| = 0$) in order to prevent the indeterminate case, where the numerator is null. On the other hand, $+1$ (corresponding to $\log_k(k)$) refers to the case, where the denominator is null; when a fusion of the two sets $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$ is found, to prevent the indeterminate case. More formally, the stability $\Psi^{t}_{w_i}$ is defined as follows:

$$\Psi^{t}_{w_i} = \begin{cases} \log_k(\frac{1}{k}), & \text{if } S^{t}_{w_i} \cap S^{t-1}_{w_i} = \emptyset \\ \log_k(k), & \text{if } S^{t}_{w_i} = S^{t-1}_{w_i} = k \\ \textit{Equation 7.2}, & \text{otherwise} \end{cases} \tag{7.3}$$

After computing $\Psi^{t}_{w_i}$ corresponding to each keyword of interest w_i over all timespans T, the average of stability Ψ^{t} of n selected keywords of interest is computed, where $n \leq |V_t|$ at a timespan t as follows:

$$\Psi^{t} = \frac{\sum_{i=1}^{n} \Psi^{t}_{w_i}}{n} \tag{7.4}$$

Interpretation

The computation of k-NN stability is based on the dynamism of the keywords appearing and disappearing in $S^{t-1}_{w_i}$ and $S^{t}_{w_i}$. Four types of keywords can be defined based on their dynamism: *recurrent keywords*, *non-recurrent keywords*, *persistent keywords*, *emerging keywords* and *dying keywords*.

Definition 1 *(Recurrent keyword) A word w_j is called* recurrent *if it appears recurrently in the k-NN subsequent sets $S^{t}_{w_i}$ and $S^{t_r}_{w_i}$, $t_r \geq 2$.*

Definition 2 *(Non-Recurrent keyword) Contrary to* recurrent keyword, *a* non-recurrent *keyword does not appear in the k-NN subsequent sets; it appears in $S^{t}_{w_i}$, but never appears in $S^{t_r}_{w_i}$, $t_r \geq 2$.*

Definition 3 *(Persistent keyword) A word w_j is* persistent *if it appears in $S^t_{w_i}$ and at least in $S^{t+1}_{w_i}$.*

Definition 4 *(Emerging keyword) A word w_j is called* emerging *if it appears in the k-NN set $S^{t+1}_{w_i} \setminus S^t_{w_i}$.*

Definition 5 *(Dying keyword) A word w_j is called* dying *if it it appears in the k-NN set $S^t_{w_i} \setminus S^{t+1}_{w_i}$.*

These definitions are used to provide a fine-grained analysis of the main streams of keywords based on their appearance/disappearance and frequency of appearance. This helps to fully understand the evolution of scientific keywords over time.

7.2 Experiments

To track the dynamics of the scientific literature in the domain of machine learning, the proposed approach *Vec2Dynamics* is evaluated on a time-stamped text corpora extracted from the NIPS conference proceedings. The experiments in this chapter demonstrate that the proposed approach delves into the content of research papers and provides a deep descriptive analysis of the literature of machine learning over 29 years by following temporal embeddings and analysing the resulting dynamic k-NN keywords over time.

7.2.1 NIPS Dataset

The dataset used in this analysis represents 29 years of the NIPS conference papers published between 1987 and 2016, with a total of 6562 papers. The dataset was first preprocessed following the steps described in Section 7.1.1. Then, in order to study the dynamics of the scientific literature of machine learning by tracking the k-NN of keywords/topics over time, the NIPS publications between 1987 and 2016 are divided into ten 3-year timespans. The length of the timespan is based on the study performed by Anderson *et al.* (Ashton et al., 2012) on evolving scientific topics. Their investigations showed that the interval of three years was successful to track the flow of scientific corpora. The statistics of the dataset are given in Table 7.1.

TABLE 7.1: Statistics of the NIPS dataset (1987 – 2016)

timespan	♯Papers	♯Words	♯Vocabulary
From 1987 to 1989	288	16,273	9147
From 1990 to 1992	417	465,169	169,728
From 1993 to 1995	453	914,871	1,669,54
From 1996 to 1998	456	1,387,070	173,341
From 1999 to 2001	499	1,943,821	197,845
From 2002 to 2004	615	2,716,271	264,241
From 2005 to 2007	631	3,595,398	292,681
From 2008 to 2010	807	4,847,535	379,086
From 2011 to 2013	1037	6,501,435	480,440
From 2014 to 2016	1386	8,732,443	610,383

Table 7.1 shows a positive trend in the evolution of the number of papers per 3-years over the 1987–2016 study period. The average 3-annual growth rate is around 20%, exceeding 33% in the timespan 2014–2016. The findings revealed that there is a potential or possible emerging research keywords/topics within the new evolving papers. This could be justified by the constant growth rate of unique words that reaches 29.52% in the timespan 2008–2010, with an average rate of 17.98% overall, excluding the first timespan 1987–1989, where the vocabulary size was very small and may bias the result.

7.2.2 Results and Discussion

Vec2Dynamics is evaluated on tracking the dynamics of machine learning literature. To do so, temporal word embeddings are leveraged to trace the evolution of the main streams of machine learning keywords. To this end, the NIPS publications – published between 1987 and 2016 and divided into ten 3-years timespans – have been used.

For each timespan t, a corpus P_t of all publications published during this time period is created. Then, after preprocessing as described in Section 7.2.1, the skip-gram model of word2vec is trained at every t with the embedding dimension $N = 200$ and the context window $= 6$. The choice of these hyperparameters is supported by the previous findings detailed in Chapter 4 that showed that these hyperparameters are optimal within the NIPS corpora.

After each training at timespan t, the similarities of *keywords of interest* are computed following Equation 7.1. The keywords of interest correspond to the top 100 bigrams extracted from the titles of the publications (Dridi et al., 2019b). From these 100 bigrams, only the bigrams that appeared in the highest two levels of the *Computer Science Ontology (CSO)*[2] (Osborne and Motta, 2012) were kept . This is justified by the aim to keep the keywords as generic as possible reflecting the topics rather than the fine-grained sub-topics and detailed techniques that the ontology illustrates. This restricted the keywords of interest to only 20 bigrams.

At every timespan t and for each keyword of interest $w_i^* \in V_t$, the k-NN were selected based on cosine similarity. In this study, k is set to 10. This choice is motivated by the study performed by Hall et *al.* (Hall et al., 2008), on the history trends in *computational linguistics*. Their investigation showed that the set of 10 words was successful to represent each topic.

To select the 10-NN, two steps were followed: first, the top 300 similar keywords returned by cosine similarity were taken, and then from these 300 keywords the ones that belong to the first or the second level of the CSO were filtered out, aiming to keep the keywords as generic as possible reflecting the topics rather than the fine-grained sub-topics. This aim justifies the choice of 300 neighboring keywords at the beginning. Indeed, this value was chosen experimentally; different k values in the interval $\{50, 100, 200, 300\}$ were taken. Among these values, only 300 guaranteed the existence of at least 10 nearest neighbors keywords belonging to CSO. Recall that the choice of k values was limited by the satisfaction of a similarity threshold to fulfill in order to keep the keywords as close as possible. The choice of this threshold, which is 0.37, was based on the work done by Orkphol and Yung (Orkphol and Yang, 2019) on cosine similarity threshold with word2vec.

[2] http://skm.kmi.open.ac.uk/cso/

After defining the k-NN of each keyword of interest w_i^* at every timespan t, the k-NN stability was computed. It corresponds to the logarithmic ratio of (a) the intersection between the two sets of 10-NN of w_i^* over two subsequent timespans to (b) the symmetric difference between them as given per Equation 7.2. Therefore, the k-NN stability of w_i^* at a timespan t corresponds to the output of Equation 7.2 with the two sets of k-NN of w_i^* at t and $(t-1)$ as inputs. For instance, the k-NN stability of w_i^* at the timespan 2002–2004 describes how the k-NN of w_i^* changed between the timespan 1999–2001 and the timespan 2002–2004.

Table 7.2 shows the k-NN stability of the top 20 bigrams studied in this chapter, as well as the average stability per timespan as defined in Equation 7.4. The table does not show the results related to the first two timespans 1987–1989 and 1990–1992 because (i) the first window does not contain the studied keywords of interest, given that the vocabulary size is too small, which justifies the difficulty to build bigrams; (ii) the second window refers to the window $(t-1)$ that is used to measure the stability at the timespan 1993–1995. The overall results reveal that the k-NN stability was mostly negative. For instance, the average stability ranges from -0.24 to -0.531 (which is not surprisingly) tracing the amount of disruption in the field of machine learning. The lowest stability was detected in the time period 1996–1998. This suggests that the field may have been more innovative and receptive of new topics/keywords at the time. This is indeed supported by the *timeline of machine learning*[3] that shows an interesting amount of achievements and discoveries in the field during the period of 1995 until 1998 such as the discovery of *Random Forest Algorithm* (Ho, 1995), *Support Vector Machine* (Cortes and Vapnik, 1995), *Long-Short-Term Memory (LSTM)* (Hochreiter and Schmidhuber, 1997) and the achievement of *IBM Deep Blue* in the world champion at chess (Campbell et al., 2002). However, the highest stability was shown in the timespan 2014–2016. At that time, it seemed that the field has reached a certain maturity, which makes the k-NN stability high. To see if this was in fact the case, the k-NN of all keywords of interest were analysed at this time period. Interestingly, it was found that 70% of keywords share one of the following nearest neighbors keywords {*"neural_network"*, *"deep_learning"*, *"big_data"*}. The shift toward *big data analytics* and *deep learning* is well known in the field in the last years of the analysis. For instance, all discoveries in this period are founded on *deep neural networks* and applied to *big data* such as *Google Brain* (2012) (Le et al., 2012), *AlexNet* (2012) (Krizhevsky et al., 2012), *DeepFace* (2014) (Taigman et al., 2014), *ResNet* (2015) (K. He et al., 2015) and *U-Net* (2015) (Olaf et al., 2015).

For better interpretation of the rise and decline of k-NN stability over time, Fig 7.2 illustrates the average of k-NN stability over eight timespans. The sharp increase in stability is readily apparent in the last timespan as discussed above. On the other hand, it can be seen the decrease in stability in the period 1996–1998 and then in the period 2008–2010. The former is interpreted previously. However, the later refers to a timespan that represented an important time frame for *ImageNet* (Deng et al., 2009) that was the catalyst for the AI boom of the 21^{st} century. This may justify the decline in stability at that time.

Regarding the time period from 1999 until 2007, Fig 7.2 shows a steady k-NN stability. This suggests that the field was stable at that time; it was broadly exploring and applying what has been discovered in the disrupted period 1993–1995. This suggestion is confirmed by machine learning timeline that does not report any topics or discoveries having been prominent at that time except the release of *Torch* (Collobert

[3] https://en.wikipedia.org/wiki/Timeline_of_machine_learning

TABLE 7.2: k-NN stability of top 20 bigrams, n/a means that the keyword does not exist either in this timespan or in the previous timespan

Keywords	93-95	96-98	99-01	02-04	05-07	08-10	11-13	14-16
machine_learning	n/a	−0.6	−0.95	−0.6	−0.6	0	0.176	0
neural_network	−0.6	−0.6	−0.18	−0.95	0	−0.368	−0.18	0
supervised_learning	−0.95	−1	−1	−0.368	−0.368	−0.368	−0.368	−0.18
unsupervised_learning	−0.18	−0.6	0	−0.95	−0.6	−0.368	−0.368	−0.6
reinforcement_learning	−1	0	−0.368	−0.368	−0.18	−0.368	0	0
time_series	−0.48	−0.95	−0.95	−1	−1	−1	−1	−0.95
artificial_intelligence	n/a	n/a	n/a	n/a	−0.3	0	n/a	−0.368
gaussian_process	n/a	n/a	0	0	0	−0.18	−0.368	0.176
semi_supervised	n/a	n/a	n/a	n/a	−0.18	−0.6	−0.6	−0.6
active_learning	n/a	n/a	n/a	−0.18	−1	−1	−1	−1
decision_trees	n/a	−0.6	−0.6	−0.545	−0.95	−0.368	−0.95	0.176
dimensionality_reduction	n/a	n/a	−1	−0.18	−0.368	−0.368	−0.95	−0.6
dynamic_programming	−0.95	0	0.176	−0.368	0	−0.6	−0.6	0
gradient_descent	0.176	−0.18	−0.368	0	−0.368	−0.18	0.602	0.176
hidden_markov	0	−0.6	−0.6	−0.18	−0.18	−0.95	−0.368	−0.368
mutual_information	−0.95	−0.397	−1	−0.6	−1	−0.368	n/a	−0.552
nearest_neighbor	−0.18	−0.18	−0.18	−0.18	−0.18	−0.368	−0.18	0
pattern_recognition	−0.18	−0.95	−0.368	−0.18	n/a	n/a	−0.6	n/a
monte_carlo	−0.6	−0.6	−0.18	−0.18	0.176	0	−0.18	0
graphical_model	n/a	n/a	n/a	0.176	−0.18	−0.368	−0.18	0
Average	−0.491	**-0.531**	−0.474	−0.392	−0.38	**-0.41**	−0.395	−0.24

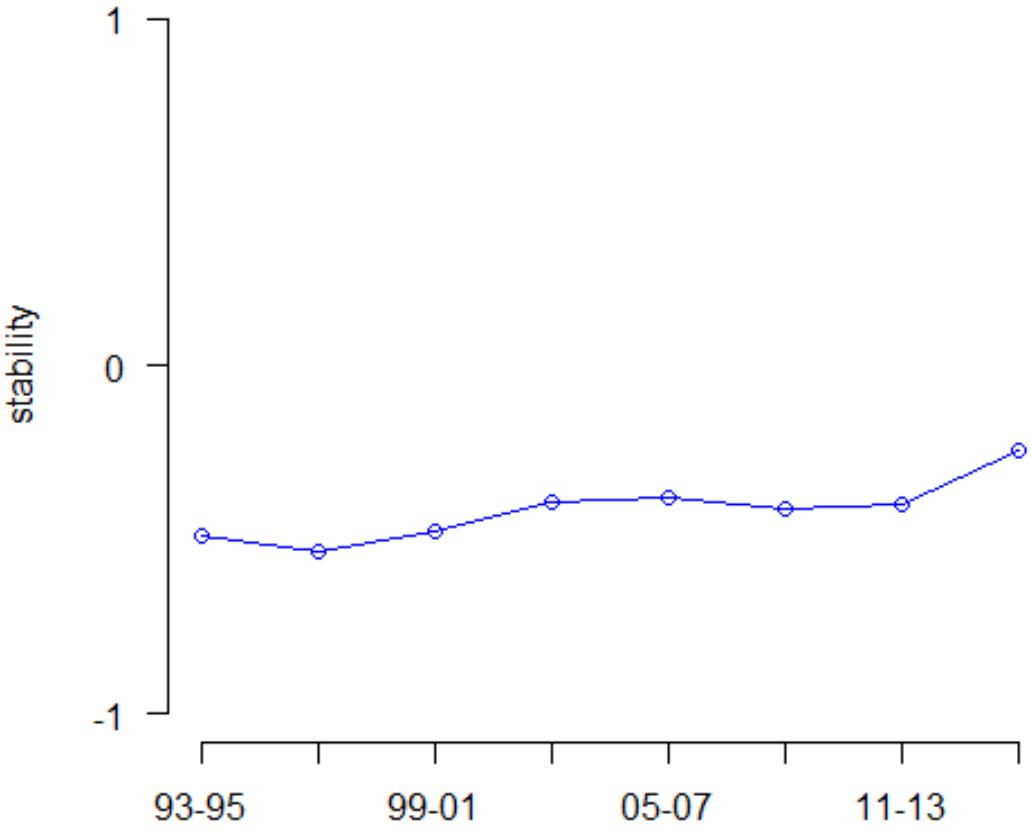


FIGURE 7.2: *k*-NN average stability over time

et al., 2002), which is actually a software library for machine learning; it is indeed a tool and does not have anything to do with new topics.

For overall results on the NIPS publications, *Vec2Dynamics* shows promising findings in tracking the dynamics of machine learning literature. As a proof of evidence, *Vec2Dynamics*, applied to the keyword of interest *"machine_learning"*, detects interesting patterns as shown in Fig 7.3 – Fig 7.9. Each figure depicts the Venn diagram of two subsequent sets of *k*-NN keywords of the keyword *"machine_learning"*. As these figures show, *"machine learning"* seems to have been stabilised significantly over time. For instance, the overlap of the two sets $S^{t-1}_{machine\_learning}$ and $S^{t}_{machine\_learning}$ has increased gradually to pass from only one keyword between the sets of the time periods 1996–1998 and 1999–2001 (Figure 7.4) to six keywords between the sets of the time periods 2008–2010 and 2011–2013 (Figure 7.8).

In addition, the fine-grained analysis of the main streams of keywords reveals the different types of keywords based on their appearance/disappearance and their frequency of appearance in the subsequent sets of *k*-NN. For instance, in the case of *"machine_learning"*, the keywords *"computer_vision"*, *"bioninformatics"*, *"robotics"* and *"economics"* for example are *recurrent* as they appeared recurrently in more than four timespans. However, the keyword *"html"* is *non-recurrent* because it appeared only in two timespans and then disappeared. On the other hand, *"nlp"* is considered an *emerging* keyword in the timespan 2002–2004 due to its first appearance. This is insightful because the field of *natural language processing* has seen a significant progress after the vast quantities of text flooding the World Wide Web in the

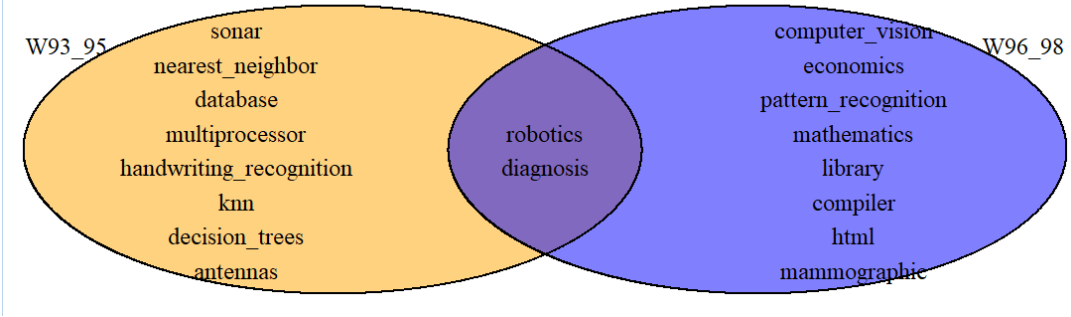


FIGURE 7.3: Venn Diagram of "machine_learning" in the timespan 1996-1998

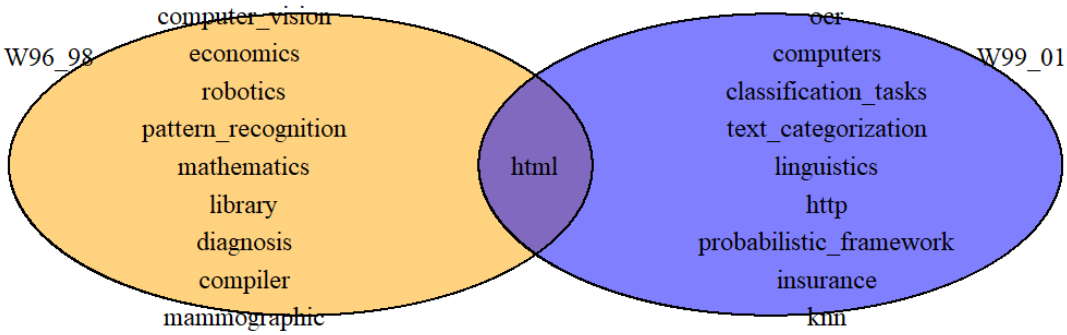


FIGURE 7.4: Venn Diagram of "machine_learning" in the timespan 1999-2001

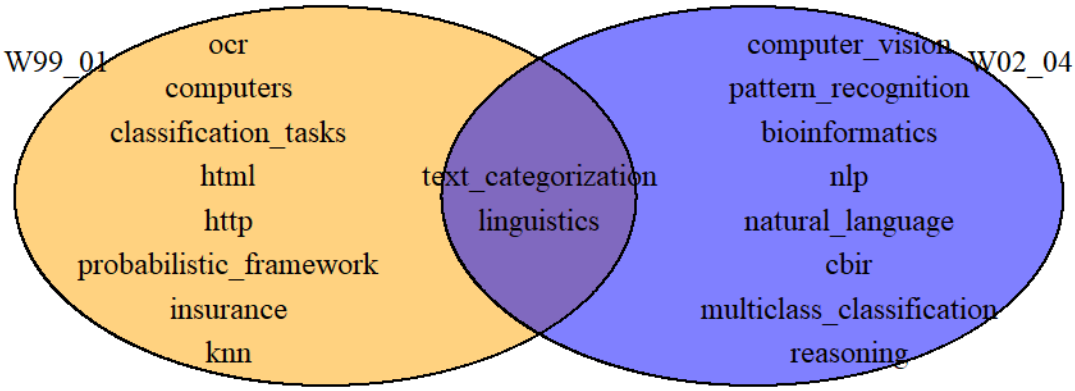


FIGURE 7.5: Venn Diagram of "machine_learning" in the timespan 2002-2004

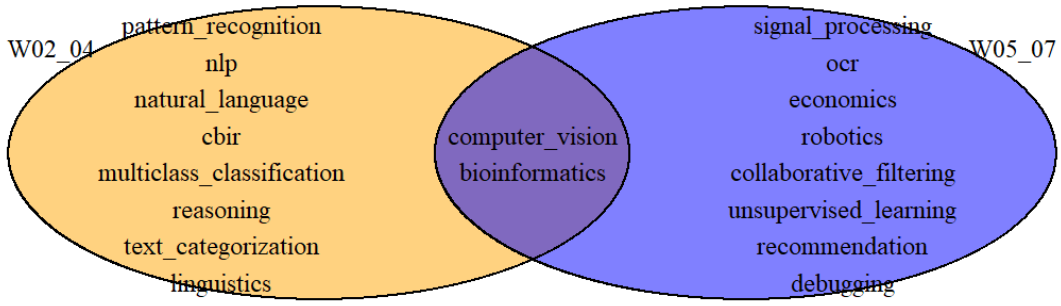


FIGURE 7.6: Venn Diagram of "machine_learning" in the timespan 2005-2007

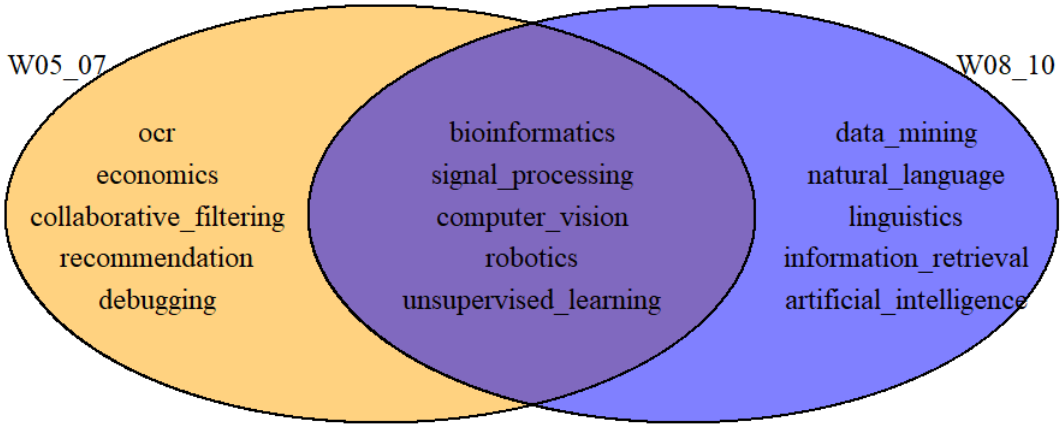


FIGURE 7.7: Venn Diagram of "machine_learning" in the timespan 2008-2010

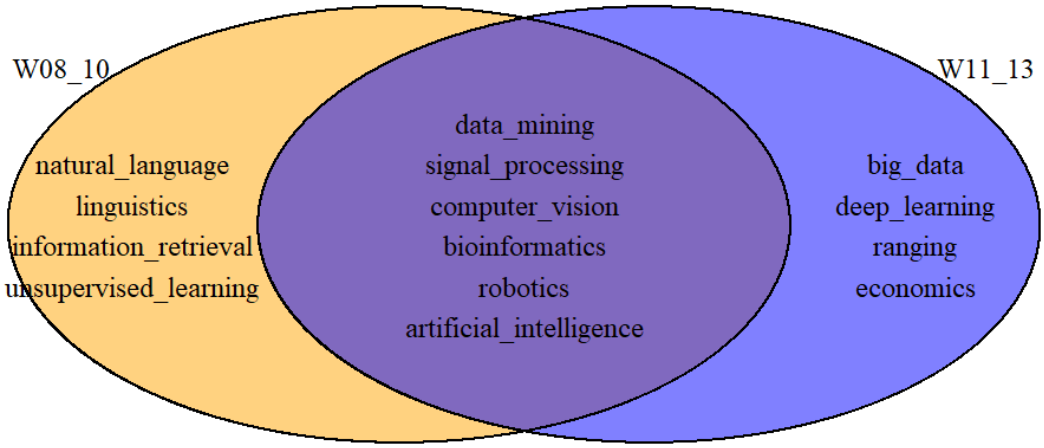


FIGURE 7.8: Venn Diagram of "machine_learning" in the timespan 2011-2013

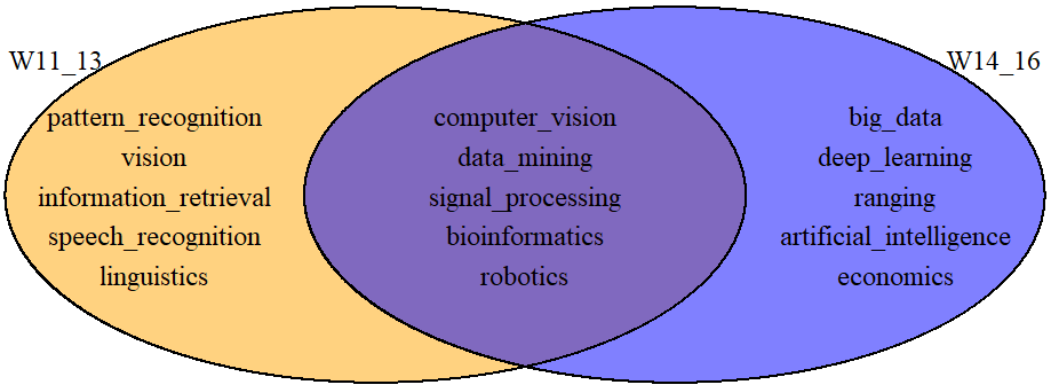


FIGURE 7.9: Venn Diagram of "machine_learning" in the timespan 2014-2016

late 1990s, notably by information extraction and automatic summarising (Inderjeet, 1999). However, the keyword *"mathematics"* is considered a *dying* keyword as it completely disappeared after the time period 1996–1998. This could be justified by the fact that early machine learning approaches and algorithms were developed based on mathematical foundations such as *Bayes' theorem*, *Markov chains*, *Least Squares*, *etc*; that is why early machine learning researchers have extensively investigated mathematics in their literature comparing to the present ones that focus more on the applications.

Overall, *Vec2Dynamics* shows a great potential to track and explore the dynamics of machine learning keywords over time. Both numerical and visual analyses show the effectiveness of the proposed approach to trace the history of machine learning literature exactly as machine learning timeline does.

7.3 Summary

This chapter represents the second path that this thesis follows to perform the computational history of science by tracking the dynamism of scientific keywords over time. *Vec2Dynamics* has been then proposed, a new temporal word embedding approach to analyse and explore the dynamics of the scientific keywords. To this end, *Vec2Dynamics* followed an innovative way by leveraging word2vec embeddings to delve into the paper content and track the dynamics of k-nearest neighbors (k-NN) keywords of a keyword of interest. To do so, the proposed approach trained temporal embeddings over ten 3-years timespans. Then, after each training, it computed the similarities between pairs of keywords of interest and accordingly it defined the k-NN keywords of each keyword of interest. Afterward, it computed the stability of k-NN over every two subsequent timespans.

Vec2Dynamics has been applied to the area of machine learning, and it has shown both numerical and visual evidences to track the dynamism of scientific keywords. A research area with growing k-NN stability is likely to subsequently gain maturity, while the contrary is also true; it refers to an emerging area with new topics becoming more prominent.

Chapter 8

Conclusion and Future Directions

> "I write the last line, and then I write the line before that. I find myself writing backwards for a while, until I have a solid sense of how that ending sounds and feels. You have to know what your voice sounds like at the end of the story, because it tells you how to sound when you begin. "
>
> — John Irving (1942 –)

This chapter summarises the contributions of this dissertation and also gives future directions of the current work.

This dissertation focuses on the task of computational history of science, which is performed by tracking the dynamics of science and detecting the emerging scientific trends. To address this task, temporal word embeddings have been used and applied to scientific literature to closely and automatically track the change in pairwise associations between pairs of scientific keywords over time. While there has been some work in recent years on the computational history of science and trend analysis, this thesis offers a new solution that differs from the existing research in several ways. First, unlike existing approaches that have focused on one of the two tasks of the computational history of science – which are the tracking of the dynamics of science and the detection of emerging scientific trends – this thesis has performed both of them adopting a variety of heuristics analysing the pairwise associations between pairs of scientific keywords, offering consequently a comprehensive overview of the history of science. Second, for the detection of emerging trends, this thesis has differentiated between *converging keywords* and *contextualising keywords* as two types of research trends. The converging keywords refer to the pairs of keywords that converge over time leading to the synthesis of a new scientific keyword. This new scientific keyword may represent a sub-field in the area, and accordingly represent a new trend. For example, *"deep learning"* is the converging keyword between the pairs of keywords *"machine learning"* and *"neural networks"*. In this respect, *"deep learning"* is a sub-field in the area of *"machine learning"*. However, the contextualising keywords represent the pairs of keywords that frequently co-occur in the same context over time. They may refer to an hybridisation of tools or techniques in the area. A notable example of contextualising keywords is *"ensemble learning"* and *"deep learning"*, where *"deep learning"* is highly used in *"ensemble learning"* as a classifier. Third, most of the current approaches rely mainly on citation counting from papers, which have been published, and consequently find clues to topic evolvement (Zehra and Umut, 2018). While citation counts are used as indicators of emerging research topics, they

can take months or even years to reveal research trends. Also, they fail to dig into the paper content. Therefore, there is a need to shift from citation-based approaches to more fast yet accurate approaches for computational history of science that drill into the content of scholarly publications.

Following this trend, some work (Ashton et al., 2012; Bakarov et al., 2018; Li et al., 2019; Paul and Girju, 2009; Salatino et al., 2017) emerged and explored natural language processing techniques, namely *topic models*, to forecast the emergence of new research topics. While topic models intend to extract semantics by capturing document level associations between words, they fail to detect pairwise associations of keywords. This is a considerable limitation since emerging topics usually start first by an increasing closeness of keywords that may lead to a merge. This closeness is generally semantic rather than linguistic. For instance, the research topic *"deep learning"* resulted from the merge between the two keywords/topics *"machine learning"* and *"neural networks"*. Similarly, *"bioinformatics"* is the keyword that emerged from the convergence between *"biology"* and *"information engineering"*. Recently, the emerging research topic *"federated learning"* resulted from the merge between *"machine learning"* and *"decentralised data"*. This thesis makes the assumption that emerging trends are defined as a pair of fast conversing keywords. However, this is not always the case. Some emerging keywords could potentially be new terms that appear as a result other than merging existing terms. This is generally the case of names of tools, techniques or programming languages.

This thesis has overcome these deficiencies by proposing a fine-grained study of the associations between pairs of keywords. This study has focused on the change in pairwise associations between keywords over time to early detect emerging trends and closely track the dynamics of science. The next section describes the contributions that this thesis brings to the field and gives an overview of the dissertation.

8.1 Overview and Contributions

Over the past few years, the computational history of science – as a part of big scholarly data analysis (Feng et al., 2017) – has grown into a scientific research area that is increasingly being applied in different domains such as business, biomedical, and computing. The serge in interest is due to (i) the explosion of publicly available data on scholarly networks and digital libraries, and (ii) the importance of the study of scientific literature, which is continuously evolving. In fact, the recent literature is rich in dealing with the enigmatic question of the dynamics of science.

This dissertation has explored three main directions for tracking the dynamics of science and detecting the emerging scientific trends as summarised in Table 8.1.

1. This thesis has explored *word2vec* (Mikolov et al., 2013e) as one of the word embedding techniques that can be applied to represent the scientific language. A methodological approach has been proposed to study the hyperparametrisation of word embeddings and deeply understand the embedding behavior within scientific corpora (*Stage 1*). This methodology, detailed in Chapter 4, has been published in (Dridi et al., 2018).

2. Based on the outcomes of *Stage 1*, the second stage (*Stage 2*) has concerned learning word embeddings across time and using their outputs to propose three different approaches that perform the computational history of science

following two paths: (1) detecting the emerging scientific trends and (2) tracking the dynamics of science. *Hist2Vec* and *Leap2Trend* are proposed to detect the emerging scientific trends while *Vec2Dynamics* is proposed to track the dynamics of scientific keywords. All of these approaches study the change in pairwise associations between pairs of scientific keywords over time following different methodologies. These approaches have been accordingly described in Chapter 5 and published as (Dridi et al., 2019a), Chapter 6 and published as (Dridi et al., 2019b), and Chapter 7, respectively.

3. In the third stage (*Stage 3*), this thesis has provided standards to validate the results of the proposed approaches in *Stage 2*. In particular:

   (a) An analogy dataset for *machine learning* – the application area of the proposed approaches – has been created by manually curating ACM hierarchy and Wikipedia outline of machine learning. This analogy dataset is made publicly available for research at `https://github.com/AmnaKRDB/Machine-Learning-Analogies`.

   (b) Because of the absence of standards for research trends, both *Google Trends* hits and *Google Scholar* citations have been proposed as gold standards. The details of the creation and the usage of these gold standards were given in Chapter 6. The code has been made publicly available for research at `https://github.com/AmnaKRDB/Leap2Trend`.

   (c) The citation analysis has been used as a competitor with the proposed approaches to conduct a comparative study. Different strategies have been followed for the comparative study. The obtained results were promising. For instance, Spearman's positive correlation was 100% as described in Chapter 5. In addition, the precision reached 90% as given in Chapter 6.

   (d) A qualitative analysis has been adopted for a better analysis of the obtained results. Both visual and descriptive analyses have been performed in Chapters 5 and 7.

Stages	Contributions	Features
Stage 1	**A methodological approach for tuning word embedding hyper-parameters**	*k*-NN embedding stability (Dridi et al., 2018)
Stage 2	**Learning temporal word embeddings and the usage of their outputs to perform a computational history of science by detecting the emerging scientific trends and tracking the dynamics of science.**	• Hist2Vec – Detection of converging keywords (Dridi et al., 2019a) • Leap2Trend – Detection of contextualising keywords (Dridi et al., 2019b) • Vec2Dynamics – Tracking the dynamism of keywords

Stage 3	**Provide standards to validate the results**	• Creation of scientific analogy datasets • Google Trends / Google Scholar • Comparative study with citation analysis • Descriptive analysis • Visual analysis

TABLE 8.1: An overview of the contributions

8.2 Answers to Research Questions

In the following, we show how this thesis answers the research questions stated in Chapter 1, Section 1.4.

1. ***Research Question 1: How to represent the scientific text with natural language processing techniques that help to reveal the semantics and the dynamics of words over time?***

   To answer this question, three sub-questions have been answered:

   - Are word embeddings – as a natural language processing technique – (namely *word2vec* (Mikolov et al., 2013e)) able to detect semantic and syntactic analogies in scientific language?

     Outcome: Affirmative: Yes, word2vec is able to detect semantic and syntactic analogies in scientific language. A rigorous evaluation has been conducted on an analogy dataset for machine learning created from ACM hierarchy and Wikipedia outline of machine learning. The embeddings generated from the scientific corpora were able to detect interesting semantic relations in machine learning such as *"unsupervised_learning (which is a technique of machine learning) is to kmeans (k-means, which is a specific technique of machine learning) as supervised_learning (which is a technique of machine learning) is to knn (k-nearest neighbours or k-NN, which is a specific technique of machine learning)"*[1]. On the other hand, for syntactic analogies, 68% accuracy has been reached, performing higher syntactic accuracy than Mikolov's *et al.* (Mikolov et al., 2013c), which is 61%. More details were given in Chapter 4. In this thesis, word2vec has been chosen as a word embedding technique to represent the scientific language. However, any word embedding technique can be applied, and the proposed methodological approach for tuning the hyper-parameters can be used as the methodology is generalisable and can work irrespective of any word embedding technique.

[1]The keywords 'unsupervised_learning", "kmeans", 'supervised_learning" and "knn" are spelled here exactly as they are spelled in the dataset, and as they are learned by word2vec.

- How word embedding hyper-parameters can be tuned?

  Outcome: This thesis has proposed a methodological approach based on *k-NN stability* to tune the hyper-parameters of word2vec. This proposed methodology was detailed in Chapter 4.

- How to find/create analogy datasets as gold standard to validate the ability of word embeddings to detect semantic analogies from scientific text?

  Outcome: In case of absence of gold standards analogy datasets, the creation of new ones is a must. In this thesis, as machine learning is the application domain of the proposed approaches, a machine learning analogy dataset has been created from ACM hierarchy and Wikipedia outline of machine learning. This analogy dataset is made publicly available for research at `https://github.com/AmnaKRDB/Machine-Learning-Analogies`.

Outcomes of Research Question 1: This thesis has proposed to use word embeddings, namely *word2vec* (Mikolov et al., 2013e), as a natural language processing technique to represent the scientific text. Word2vec embeddings have been validated with an analogy dataset for machine learning created from ACM hierarchy and Wikipedia outline of machine learning, and it has proved its ability to detect interesting semantic and syntactic relations within the scientific language. To guarantee the accuracy of the obtained semantic and syntactic analogies, this thesis has proposed a methodological approach to tune word2vec hyper-parameters, which have a direct impact on the generated analogies. This methodology is based on k-NN stability (Chapter 4).

2. ***Research Question 2: How to explore the vector representation of words in order to study the semantic shifts of scientific keywords, and consequently perform the computational history of science?***

   To answer this question, the following sub-questions have been answered:

   - How to detect the semantic shifts of scientific keywords over time?

     Outcome: In order to track the semantic shifts of scientific keywords over time, this thesis has adopted two ways. The first way concerns the study of the similarities between pairs of keywords in order to detect their convergence over time (Chapter 5). However, the second way studies the frequent co-occurrence of pairs of keywords in the same context (Chapter 6).

   - How to perform the computational history of science?

     Outcome: In this thesis, the computational history of science was performed by following two paths. The first path detects the emerging scientific trends, which are defined in this thesis as *converging keywords* (Chapter 5) and *contextualising keywords* (Chapter 6). However, the second path tracks the dynamics of scientific keywords by studying the evolvement of their neighborhood over time (Chapter 7).

   - How to represent the temporal dimension in an effective way to perform the computational history of science?

     Outcome: The representation of the history is very important for the computational history of science. In this thesis, two different temporal

paradigms have been followed to represent the history (the temporal dimension): (i) a *static* paradigm (fixed windows of time (timespans)), and (ii) a *dynamic* paradigm: incremental and sliding windows of time. The static paradigm has been followed in Chapter 5 and Chapter 7. The length of the time windows was based on different research done on evolving scientific topics (Ashton et al., 2012) and more precisely on evolving Computer Science research (Hoonlor et al., 2013). However, in Chapter 6, the dynamic paradigm has been followed. Both incremental and sliding windows have been used. The choice of the incremental paradigm is based on the normal flow of scientific venues such as conferences and journals, which are annually publishing new papers. However, the choice of the sliding paradigm – that forgets one year vocabulary and adds one year ahead vocabulary – was motivated by the fact to keep the vocabulary as fresh as possible forgetting the old one.

Outcomes of Research Question 2: This thesis has studied the semantic shifts of scientific keywords over time by studying: (i) the pairwise similarities between keywords in order to detect the *converging keywords* (Chapter 5), and (ii) the frequent co-occurrence of pairs of keywords in the same context in order to detect the *contextualising keywords* (Chapter 6). Both the converging keywords and the contextualising keywords represent the scientific trends that this thesis defines and detects, following accordingly the first path towards the computational history of science that this thesis defines. However, the second path refers to the tracking of the dynamism of scientific keywords over time by studying the evolvement of their neighborhood (Chapter 7). In order to effectively perform the computational history of science, the temporal dimension has been represented in both static and dynamic ways in order to study the impact of the history on the evolvement of science.

3. ***Research Question 3: How to evaluate the detected emerging trends and validate the obtained results on the dynamics of science?***

   To answer this question, the following five sub-questions have been answered:

   - How to find/define gold standards related to the application areas that help to define scientific trends?

     Outcome: Trend analysis requires gold standards to validate the outcomes of the proposed approaches. Due to the absence of standards on research trends, this thesis has proposed Google Trends hits as gold standard. Due to its ability to track the popularity of pairs of various words and phrases that are typed into Google's search-box over time, it has been found that Google Trends aligns with *Leap2Trend* that tracks the closeness and the contextualisation of pairs of scientific keywords over time towards new trends. More details on how Google Trends hits have been used were given in Chapter 6.

   - Which standard validation measures can be used to assess the effectiveness of the obtained results?

     Outcome: Three standard validation measures have been used to assess the effectiveness of the detected trends, which are *precision*, *recall* and *accuracy*. All of them have been defined to answer the following two questions: (i) How accurate is the proposed approach in predicting future

trends at an early stage? (ii) How precise is the proposed approach in following the flow of Google Trends hits and citation counts?
A fine-grained methodology has been followed to define these measures in an appropriate way that aligns with the features of the proposed approach (*Leap2Trend*). This methodology has been fully described in Section 6.2, in Chapter 6.

- How to conduct comparative studies with existing approaches?
  Outcome: The two proposed approaches for trend analysis *Hist2Vec* and *Leap2Trend* have been validated with the citation counting approach, which is widely used in the literature and provides a snapshot of a fast-growing field. The objective is to check the extent to which citation analysis supports the findings of the two proposed approaches. To this end, *Spearman's correlation coefficient* has been proposed to measure the strength and direction of association between two variables that vary from one approach to another. For *Hist2Vec*, these two variables are (a) the acceleration of citation counts of publications mentioning the keywords of interest in their titles and (b) the acceleration of similarities of these keywords. However, for *Leap2Trend*, these two variables represent (a) the ascents that represent the uprankings of pairs of keywords detected by the approach and (b) any of the hits or citations returned by Google Trends hits or Google Scholar citations. More details were given in Chapter 5 and 6, respectively.
- Which standards can be defined for descriptive analysis, where normative analyses are not suitable for the analysis of the computational history of science?
  Outcome: This thesis has proposed to use the *machine learning timeline*[2] as standard for descriptive analysis. A generally good consistency between the obtained results and the machine learning timeline has been found. More details on this analysis were given in Chapter 7.
- How visual analyses can be used as qualitative analyses to highlight the semantic shifts of scientific keywords over time?
  Outcome: Both *t-SNE representations* and *Venn diagrams* have been used in this thesis for qualitative analyses to highlight the semantic shifts of scientific keywords over time. For instance, t-SNE representations have been used in Chapter 5 with *Hist2Vec* to visualise the acceleration of similarities between pairs of keywords that may lead to converging keywords. However, Venn diagrams have been used in Chapter 7 with *Vec2Dynamics* to show the evolvement of the semantic neighborhood of scientific keywords over time.

Outcomes of Research Question 3: Based on the task of the computational history of science (trend analysis or tracking the dynamics of science), the thesis has performed an adequate methodology to validate the obtained results. For instance, this thesis has addressed the challenge of lack of gold standards to evaluate the outcomes of trend analysis by building a gold standard relying on Google Trends hits. Chapter 6 has detailed how Google Trends hits align with the proposed approach *Leap2Trend* that tracks the closeness and the contextualisation of pairs of scientific keywords over time towards new trends.

[2] https://en.wikipedia.org/wiki/Timeline_of_machine_learning

Three standard validation measures have been used to assess the effectiveness of the detected trends, which are *precision*, *recall* and *accuracy*. On the other hand, in order to conduct comparative studies with existing approaches, the citation counting approach has been proposed. To this end, *Spearman's correlation coefficient* has been used to measure the strength and direction of association between the outcomes of (a) the two proposed approaches *Hist2Vec* (Chapter 5) and *Leap2Trend* (Chapter 6), and (b) the citation counts. However, on the other side, in order to evaluate the outcomes of the task of tracking the dynamics of science, both descriptive and visual analyses have been proposed. For the descriptive analyses, the machine learning timeline has been proposed as a standard (Chapter 7). However, for the visual analyses, both t-SNE representations (Chapter 5) and Venn diagrams (Chapter 7) have been used to highlight the semantic shifts of scientific keywords over time.

8.3 Future Work

There are many potential directions for future work to extend the current work in both the technical side; word embedding hyper-parametrisation, and the application side; the computational history of science. Some research directions from the current status of this work are sketched in the following:

- **Word2vec hyper-parametrisation.** This thesis has addressed word2vec hyper-parametrisation by only focusing on two hyper-parameters, which are *vector dimensionality* and *window context*. It would be interesting to assess the effects of other hyper-parameters and investigate more settings for word2vec within the scientific area.

- **Other word embedding techniques.** This thesis has used *word2vec* as a word embedding technique to represent the scientific corpora. It would be interesting to try different word embedding techniques for scientific language and compare their outputs with word2vec.

- **Other corpora and application areas.** In this thesis, the proposed approaches of the computational history of science have been applied to only the areas of machine learning and bioinformatics, and for machine learning only the NIPS corpora has been used. Further machine learning corpora could be used in the future to test the proposed methods. On the other hand, further research could usefully generalise these approaches on different research areas such as *physics*, *biology* or *medicine*, where it would be interesting to see whether a novel treatment or a certain combination of drugs for cancer is beginning to rise, as an example.

- **Other linguistic resources.** In this thesis, the proposed approaches of the computational history of science have been applied to only the scientific language resources, and English language. Further research could usefully generalise these approaches on different languages and different linguistic resources such as *press*, *history* or *forensics*. For example, for *forensics*, it would be interesting to see whether possible to predict crimes before they happen, as social scientists have long believed that historical crime trends often influence future patterns.

- **More robust gold standard.** In this thesis, Google Trends have been used as a gold standard to define trends. However, Google Trends does not cover the years before 2004, and only provides a relative search value and does not provide an exact search volume. It is suggested in future studies to consider more big scholarly data resources to fill in this gap, and provide a more robust gold standard that could be adopted to further study the task of detecting emerging scientific trends.

- **More settings of the proposed approaches.** For instance, the current version of the proposed approach *Leap2Trend* only focuses on hits in ranking to study the dynamics of research topics. This indicator may not be enough to fully understand the dynamics of science. It is thus recommended that further research be undertaken to investigate *falls* in ranking and study their impact to show the outdated research topics.

- **More linguistic analysis.** In this thesis, only unigrams and bigrams have been considered to generate the scientific vocabulary. More combinations might be explored in future work. For example, trigrams could be considered due to their significant appearance in the scientific language, such as *"social graph analysis"*, *"support vector machines"*, *"etc"*.

- **Different machine learning techniques.** This thesis has used k-NN to study the dynamics of keywords over time. Further research might explore how different clustering techniques (*e.g., Chameleon* (Karypis and Kumar, 1999)) could perform with tracking the dynamism of keywords between different clusters over time in the scientific corpora.

- **Explore social media.** This thesis was only limited to features related to research papers such as keywords and citations. However, nowadays different resources are available to enrich the set of known features such as social media. This is useful as it gives insights to investigate different resources that exist outside the realm of research papers, such as online media and social networks to detect emerging trends. This is motivated by the fact that scientists and researchers are increasingly using social media to discover new research opportunities, discuss research with colleagues and disseminate research information, which allows to track public attention and public recognition of emerging topics.

Appendix A

t-SNE Visualisations of Unigrams

In the following the t-SNE representations of the top 100 unigrams are shown (FIGURE A.1 – FIGURE A.6).

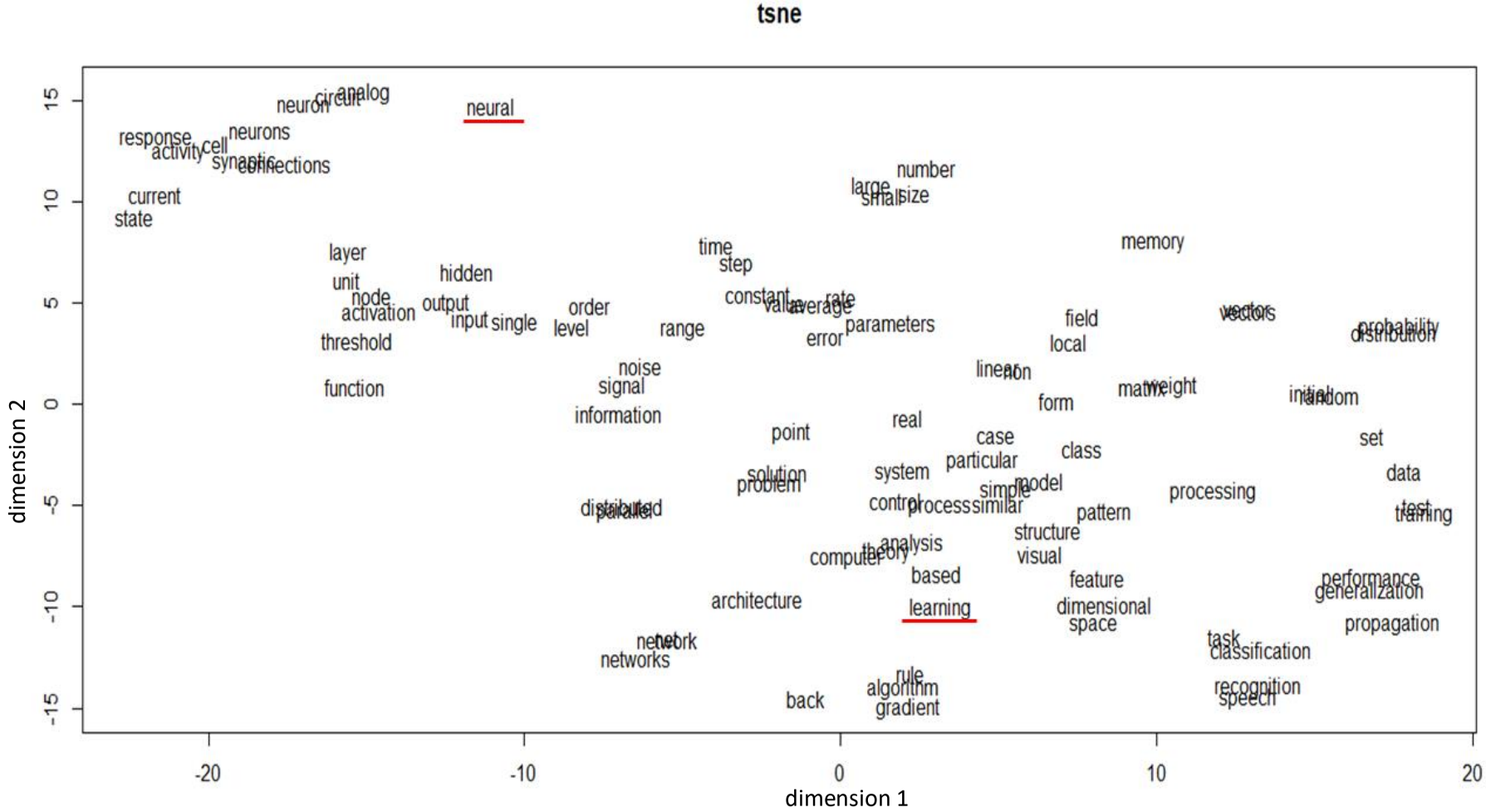


FIGURE A.1: t-SNE of top 100 unigrams of the timespan 1987-1991

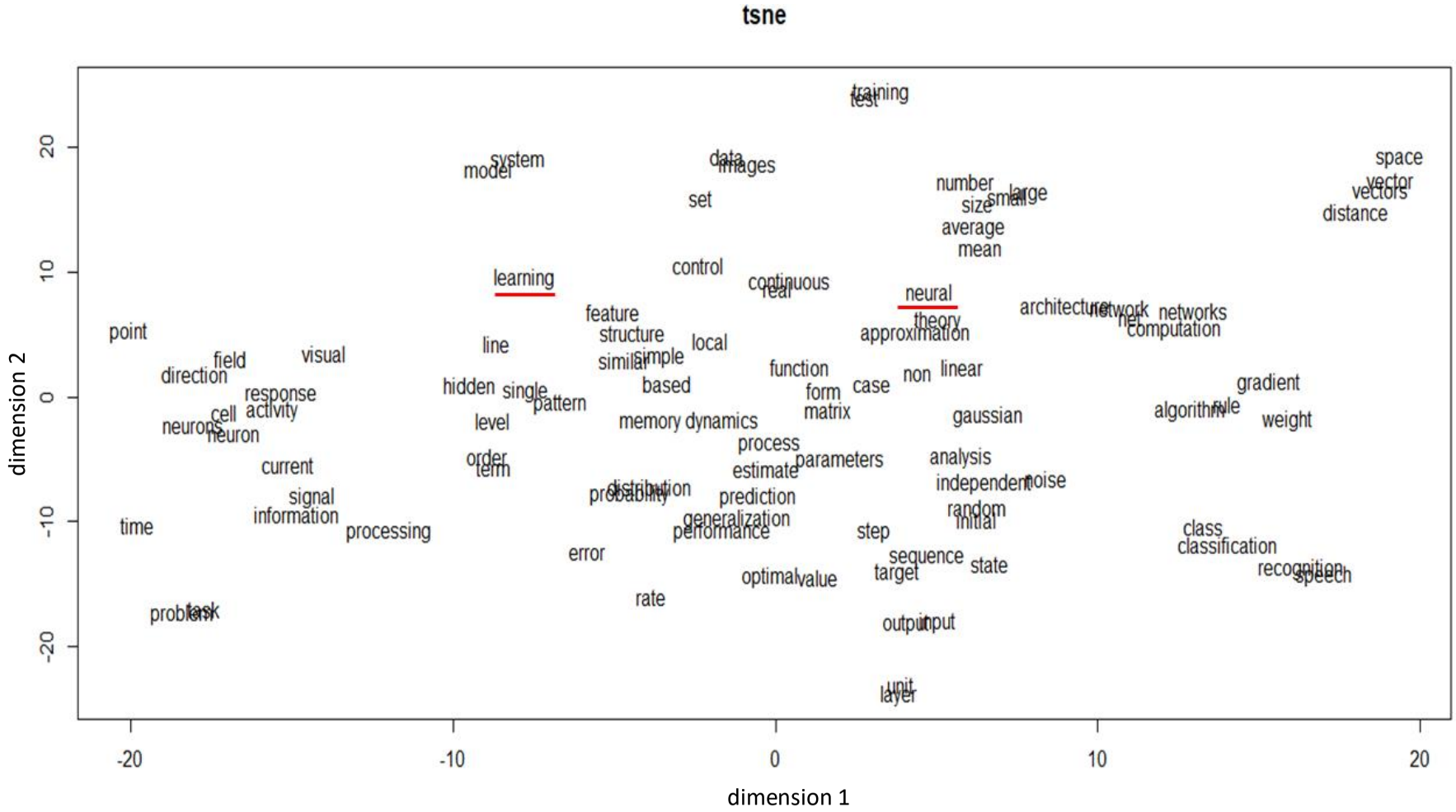


FIGURE A.2: t-SNE of top 100 unigrams of the timespan 1992-1996

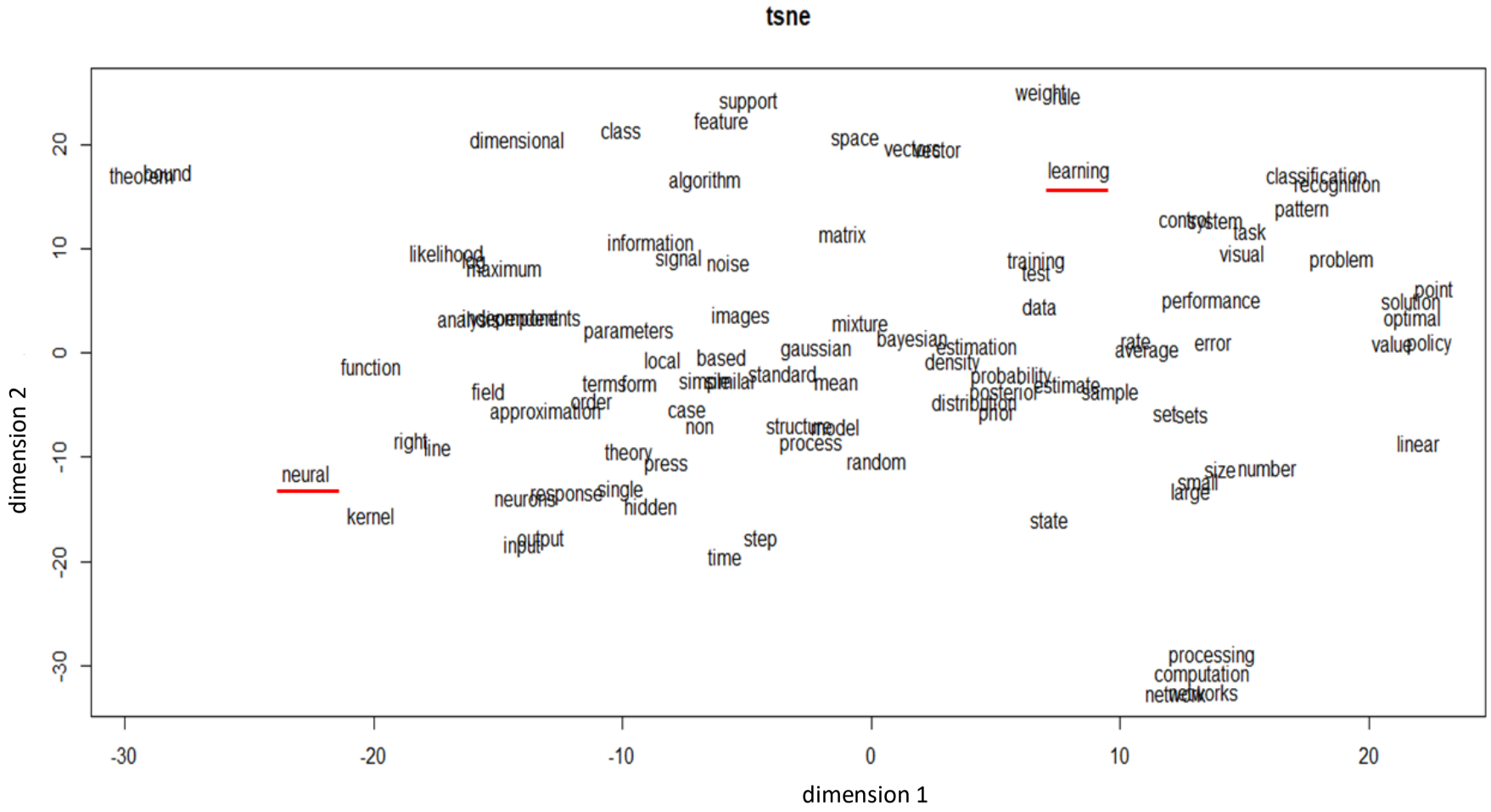


FIGURE A.3: t-SNE of top 100 unigrams of the timespan 1997-2001

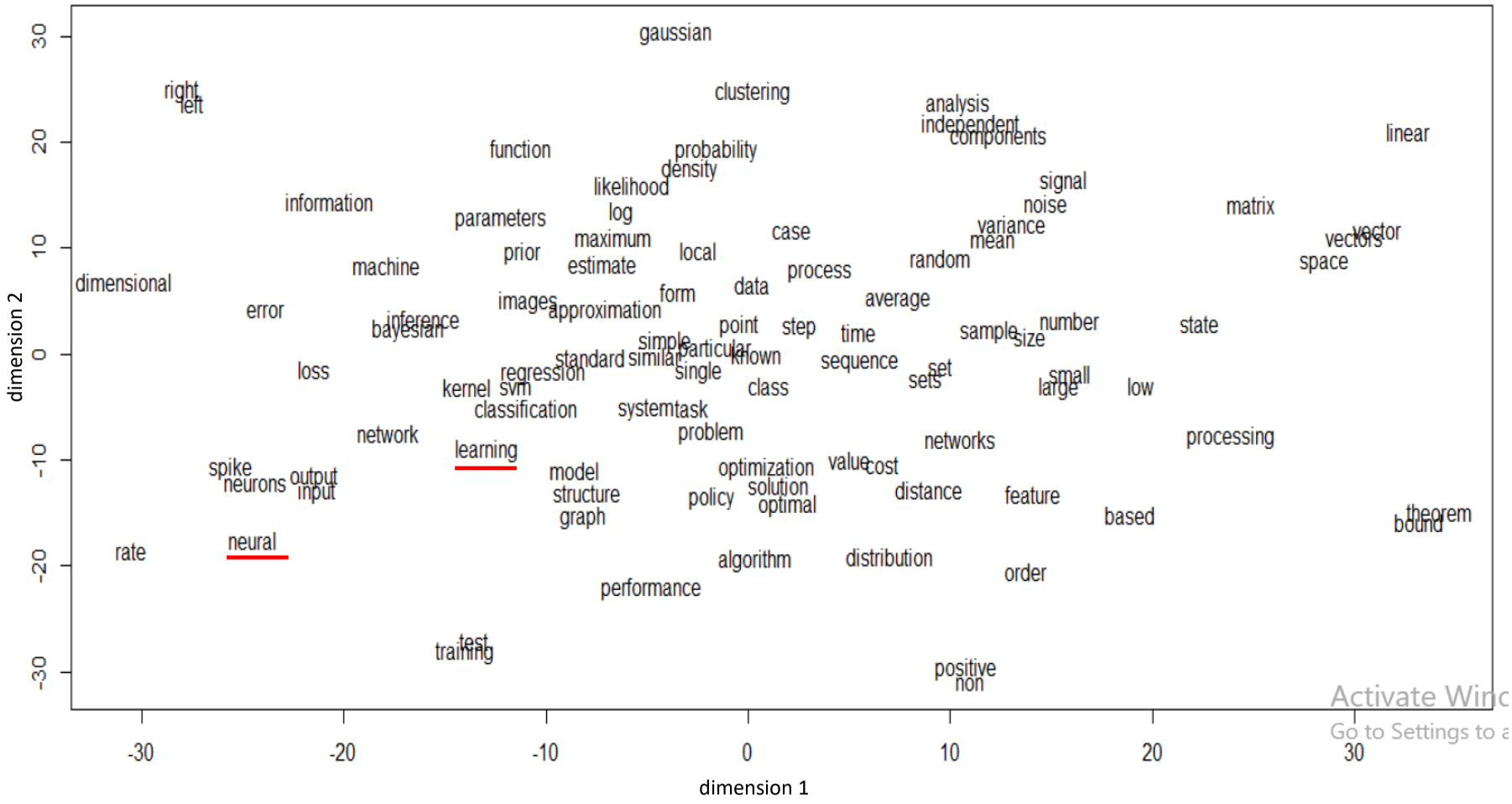


FIGURE A.4: t-SNE of top 100 unigrams of the timespan 2002-2006

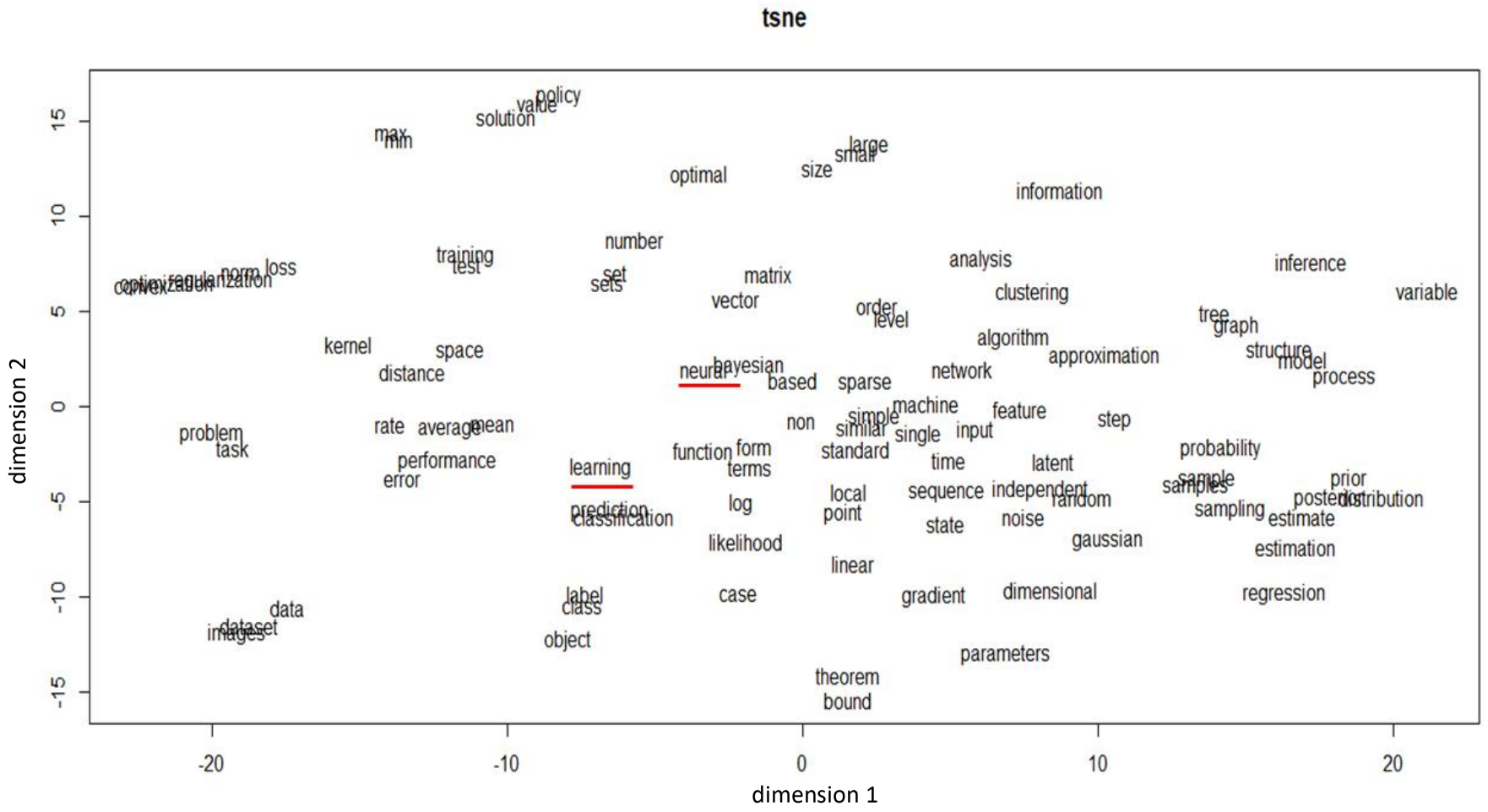


FIGURE A.5: t-SNE of top 100 unigrams of the timespan 2007-2011

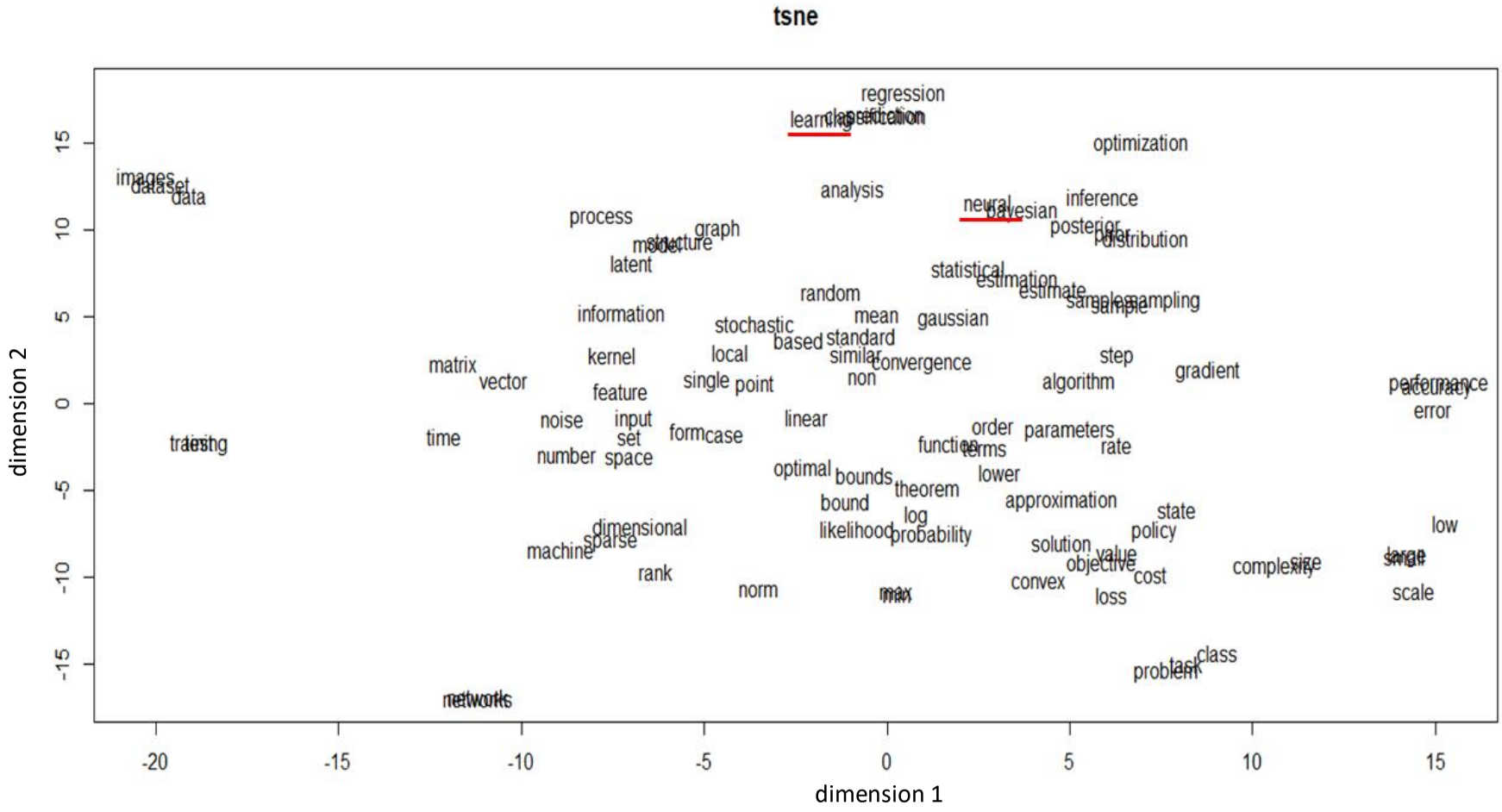


FIGURE A.6: t-SNE of top 100 unigrams of the timespan 2012-2015

Appendix B

t-SNE Visualisations of Bigrams

In the following the t-SNE representations of the top 20 bigrams are shown (Figure B.1 – Figure B.4).

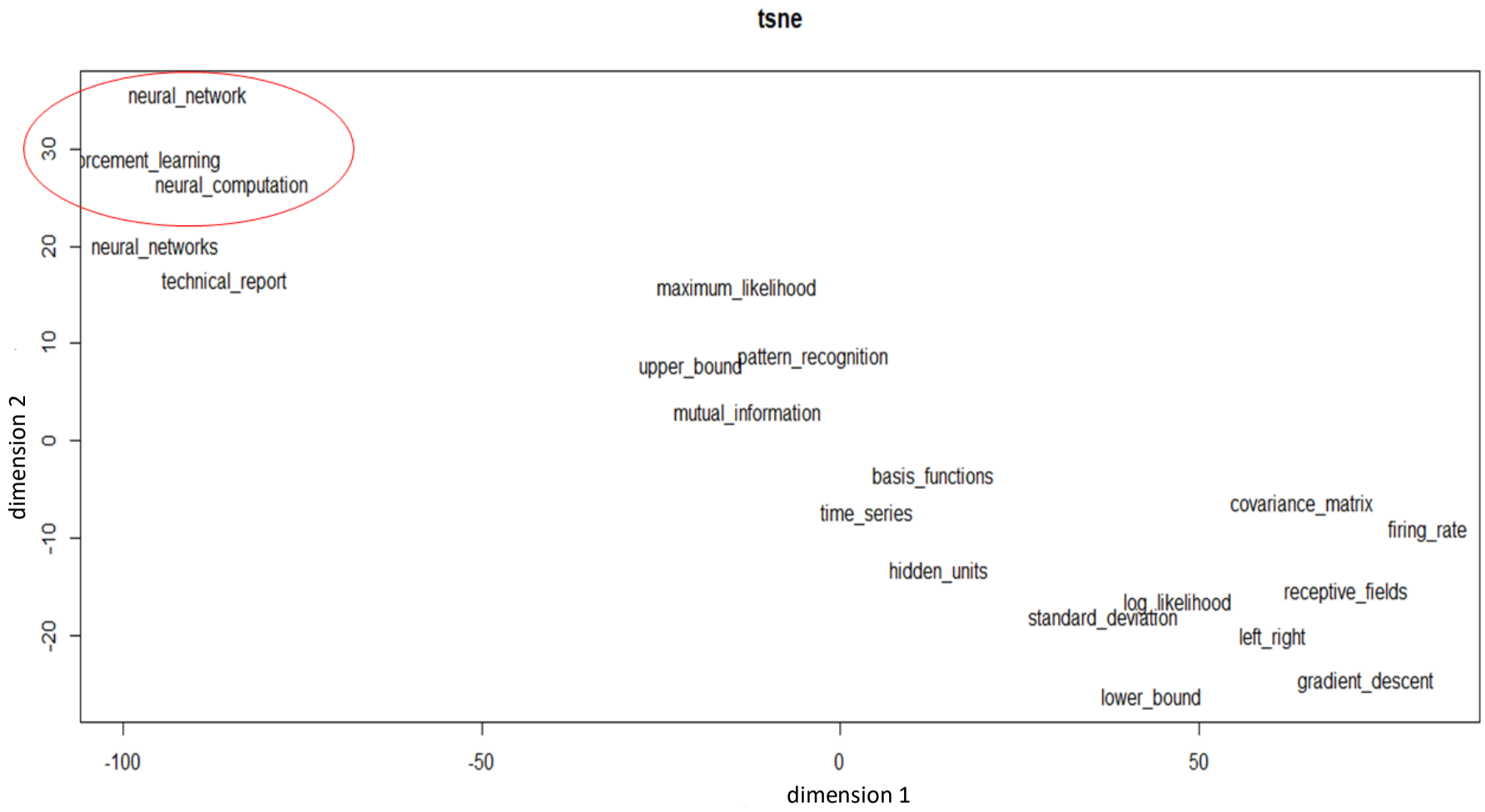


FIGURE B.1: t-SNE of top 20 bigrams of the timespan 1997-2001

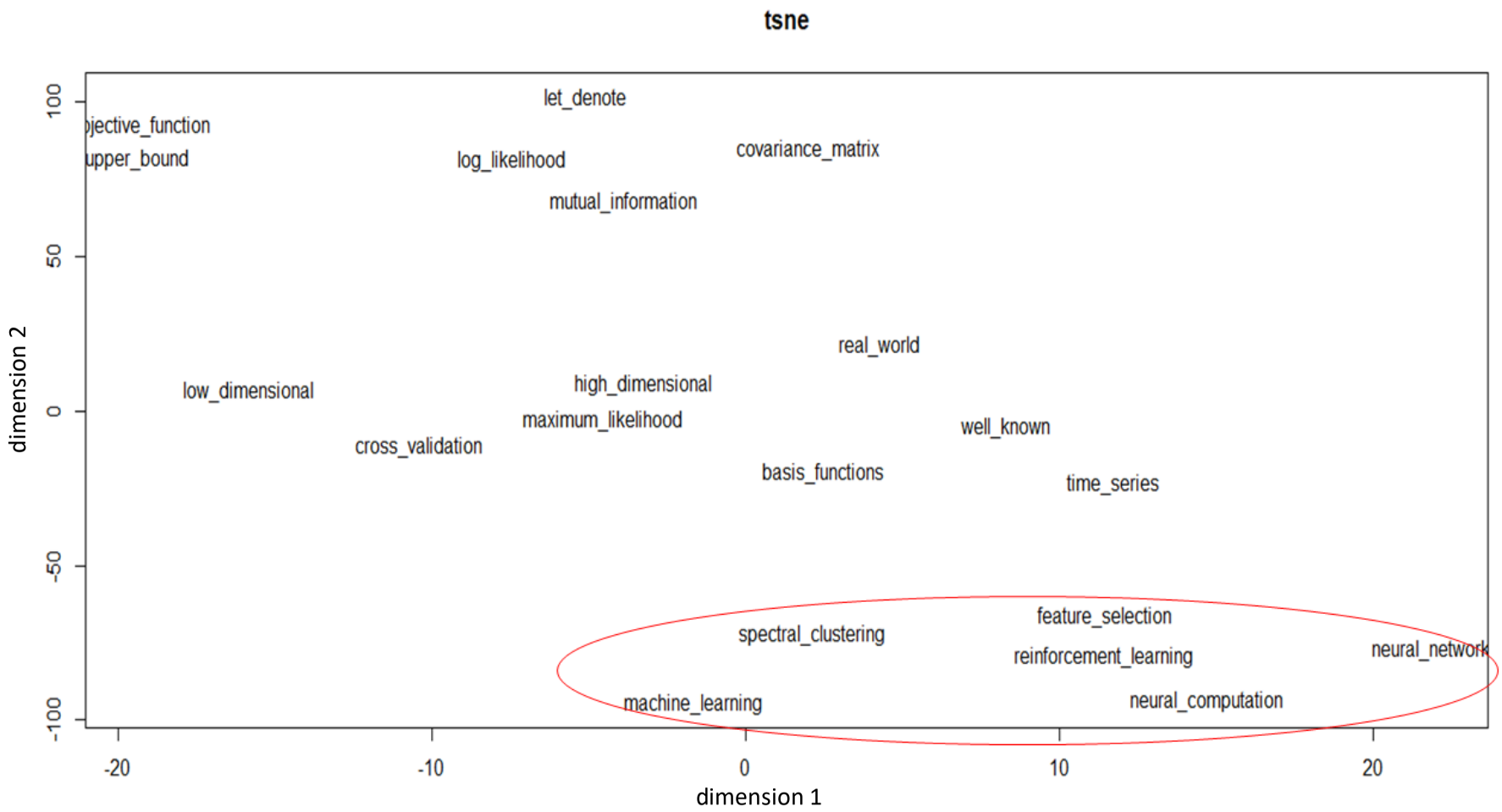


FIGURE B.2: t-SNE of top 20 bigrams of the timespan 2002-2006

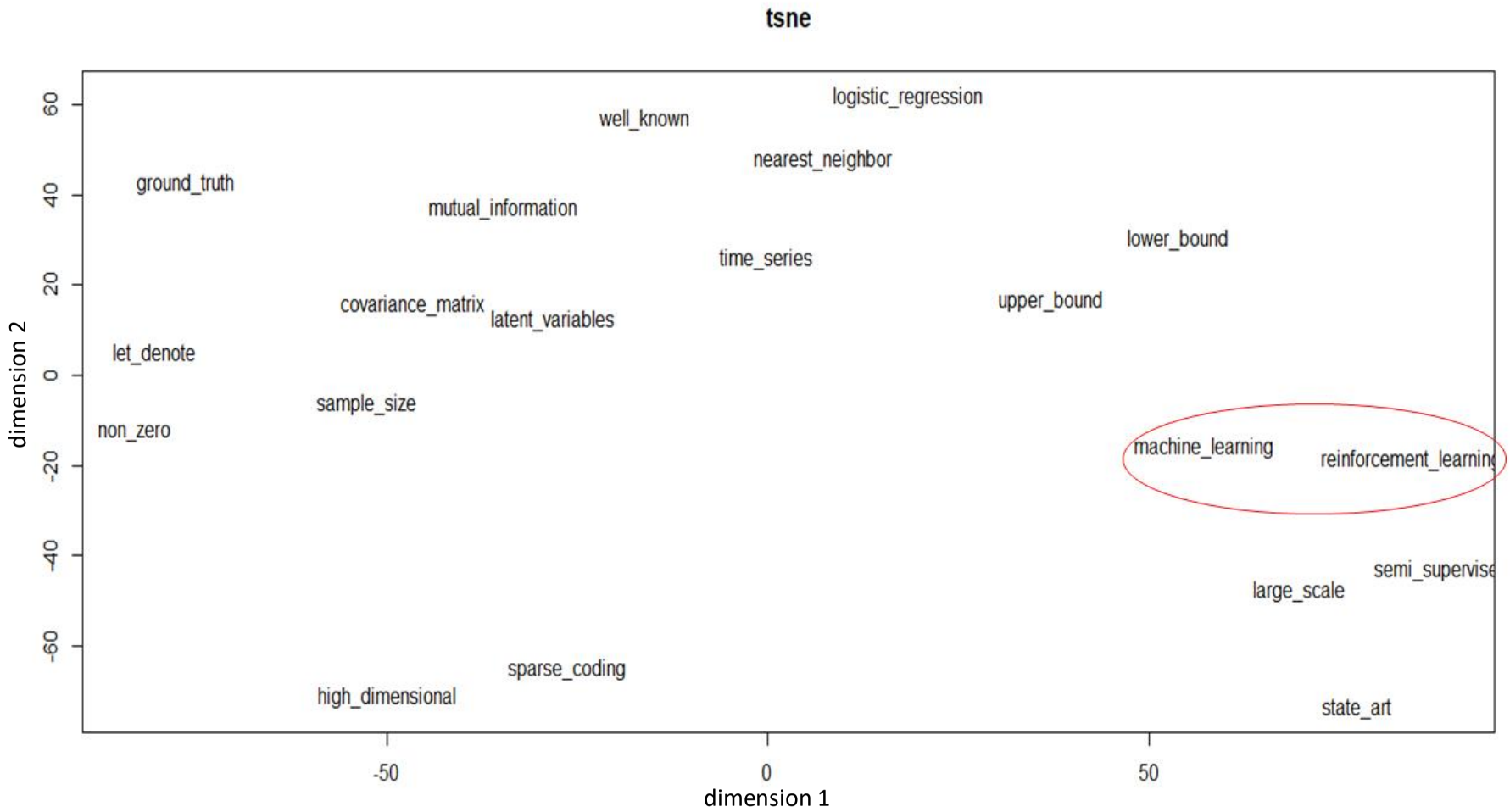


FIGURE B.3: t-SNE of top 20 bigrams of the timespan 2007-2011

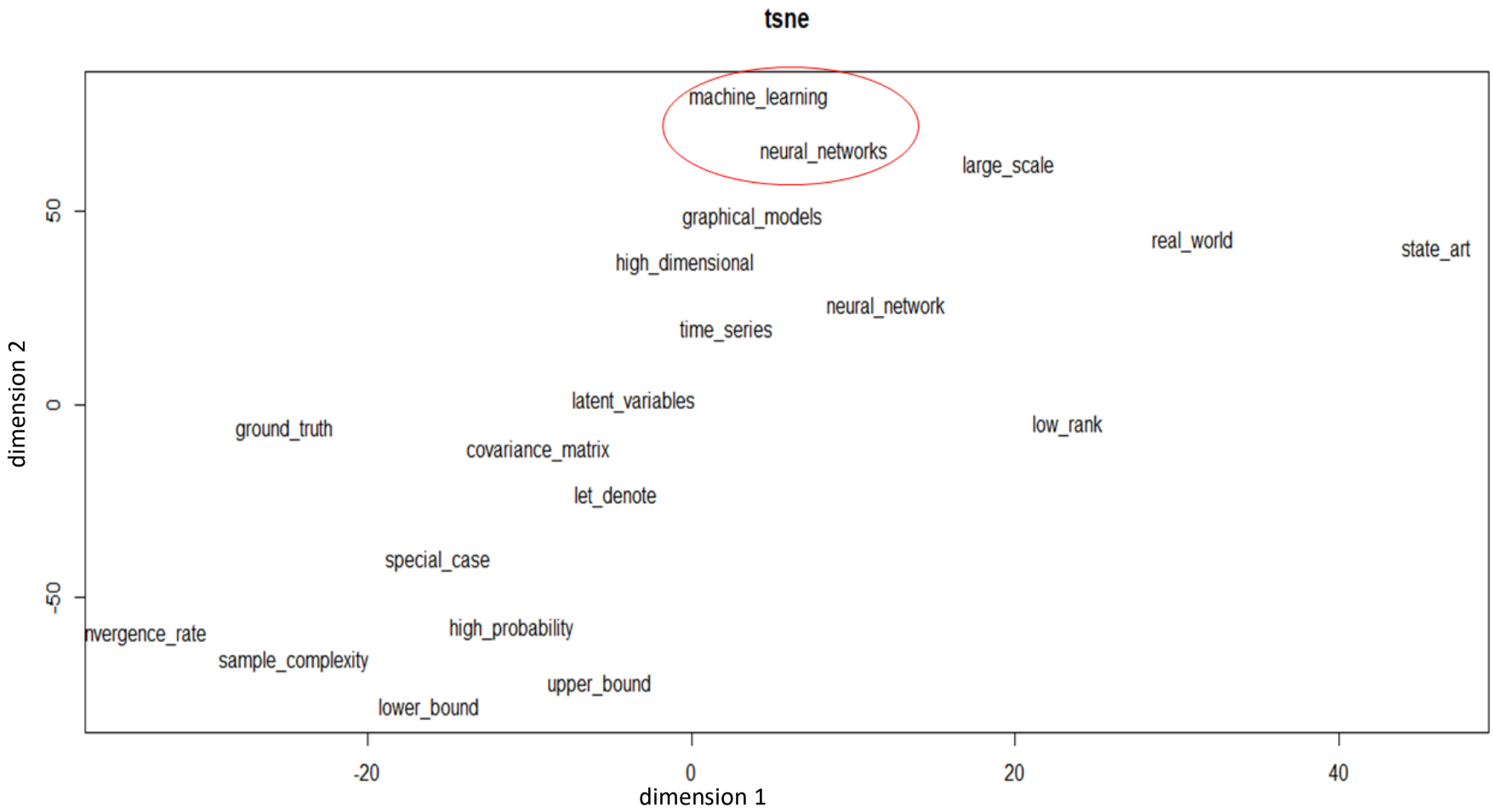


FIGURE B.4: t-SNE of top 20 bigrams of the timespan 2012-2015

c

Bibliography

Acuna, Daniel, Stefano Allesina, and Konrad Kording (Sept. 2012). "Future impact: Predicting scientific success". In: *Nature* 489, pp. 201–2.

Aduku, Kuku Joseph, Mike Thelwall, and Kayvan Kousha (2017). "Do Mendeley reader counts reflect the scholarly impact of conference papers? An investigation of computer science and engineering". In: *Scientometrics* 112, pp. 573–581.

Alam, Mohammad Mahbub and Maizatul Akmar Ismail (Dec. 2017). "RTRS: a recommender system for academic researchers". In: *Scientometrics* 113.3, pp. 1325–1348.

Alpaydin, Ethem (2010). *Introduction to Machine Learning*. 2nd. The MIT Press. ISBN: 026201243X.

Amin, Kareem et al. (2018). "Case-based Reasoning in Natural Language Processing: Word2vec VS fastText". In: *Proceedings of the 23rd UK Workshop on Case-Based Reasoning. UK Workshop on Case-Based Reasoning (UKCBR-2018), befindet sich SGAI International Conference on Artificial Intelligence, December 11-13, Cambridge, United Kingdom*. School of Computing, Engineering and Mathematics, University of Brighton, UK.

Amin, Kareem et al. (2020). "Advanced Similarity Measures Using Word Embeddings and Siamese Networks in CBR". In: *Intelligent Systems and Applications*. Ed. by Bi Yaxin, Bhatia Rahul, and Kapoor Supriya. Cham: Springer International Publishing, pp. 449–462.

An, Yoonjung, Mintak Han, and Yongtae Park (Nov. 2017). "Identifying dynamic knowledge flow patterns of business method patents with a hidden Markov model". In: *Scientometrics* 113.2, pp. 783–802.

Ashton, Anderson, Dan McFarland, and Dan Jurafsky (2012). "Towards a Computational History of the ACL: 1980-2008". In: *ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pp. 13–21.

Asooja, Kartik et al. (May 2016). "Forecasting Emerging Trends from Scientific Literature". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), pp. 417–420.

Bai, X. et al. (Jan. 2019). "Scientific Paper Recommendation: A Survey". In: *IEEE Access* 7, pp. 9324–9339.

Bakarov, Amir, Andrey Kutuzov, and Irina Nikishina (June 2018). "Russian Computational Linguistics: Topical Structure in 2007-2017 Conference Papers". In: *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2018"*, pp. 1–13.

Baroni, Marco and Alessandro Lenci (Dec. 2010). "Distributional Memory: A General Framework for Corpus-based Semantics". In: *Comput. Linguist.* 36.4, pp. 673–721.

Baum, Leonard E. and Ted Petrie (1966). "Statistical Inference For Probabilistic Functions Of Finite State Markov Chains". In: *Ann. Math. Statist.* 37.6, pp. 1554–1563.

Bengio, Yoshua et al. (2003). "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3, pp. 1137–1155.

Bhatia, Sumit and Prasenjit Mitra (Mar. 2012). "Summarizing Figures, Tables, and Algorithms in Scientific Publications to Augment Search Results". In: *ACM Trans. Inf. Syst.* 30.1, 3:1–3:24.

Bhatia, Sumit, Prasenjit Mitra, and C. Lee Giles (2010). "Finding Algorithms in Scientific Articles". In: *World Wide Web conference*. Association for Computing Machinery, pp. 1061–1062.

Blei, David M. (Apr. 2012). "Probabilistic Topic Models". In: *Commun. ACM* 55.4, pp. 77–84.

Blei, David M., NgY. Andrew, and Michael I. Jordan (2003). "Latent dirichlet allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5.1, pp. 135–146. DOI: 10.1162/tacl_a_00051.

Bornmann, Lutz and Hans Dieter Daniel (2008). "What do citation counts measure? A review of studies on citing behavior". In: *Journal of Documentation* 64.1, pp. 45–80.

Bornmann, Lutz and Robin Haunschild (May 2018). "Allegation of scientific misconduct increases Twitter attention". In: *Scientometrics* 115.2, pp. 1097–1100.

Bornmann, Lutz and Rüdiger Mutz (2014). "Growth rates of modern science: A bibliometric analysis". In: *CoRR* abs/1402.4578. arXiv: 1402.4578. URL: http://arxiv.org/abs/1402.4578.

Boser, Bernhard E., M. Isabelle Guyon, and Vladimir N. Vapnik (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, pp. 144–152.

Boyack, Kevin W., Caleb Smith, and Richard Klavans (Feb. 2018). "Toward predicting research proposal success". In: *Scientometrics* 114.2, pp. 449–461.

Boyack, Kevin W. et al. (2017). "Characterizing in-text citations in scientific articles: A large-scale analysis". In: *J. Informetrics* 12, pp. 59–73.

Burgess, Curt (June 1998). "From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model". In: *Behavior Research Methods, Instruments, and Computers* 30, pp. 188–198.

Campbell, Murray, Jr. A. Joseph Hoane, and Feng-hsiung Hsu (2002). "Deep Blue". In: *Artif. Intell.* 134.1-2, pp. 57–83.

Carlo, Valerio Di, Federico Bianchi, and Matteo Palmonari (2019). "Training Temporal Word Embeddings with a Compass". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pp. 6326–6334.

Chandra, G. and S. K. Dwivedi (2014). "A Literature Survey on Various Approaches of Word Sense Disambiguation". In: *2014 2nd International Symposium on Computational and Business Intelligence*, pp. 106–109.

Collobert, R., S. Bengio, and J. Mariéthoz (2002). *Torch: a modular machine learning software library*. Technical Report IDIAP-RR 02-46. IDIAP.

Cornelia, Caragea, Bulgarov Florin, and Mihalcea Rada (Sept. 2015). "Co-Training for Topic Classification of Scholarly Data". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 2357–2366.

Correia, Maria Joana, Isabel Trancoso, and Bhiksha Raj (2016). "Detecting Psychological Distress in Adults Through Transcriptions of Clinical Interviews". In: *Advances in Speech and Language Technologies for Iberian Languages - Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016, Proceedings*. Ed. by Alberto Abad et al. Vol. 10077. Lecture Notes in Computer Science, pp. 162–171. DOI: 10.1007/978-3-319-49169-1_16.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-Vector Networks". In: *Mach. Learn.* 20.3, pp. 273–297.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database." In: *CVPR*. IEEE Computer Society, pp. 248–255.

Deo, Narsingh (1974). *Graph Theory with Applications to Engineering and Computer Science (Prentice Hall Series in Automatic Computation)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0133634736.

Devlin, Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Dey, Ratnadeep et al. (Dec. 2017). "Sleeping beauties in Computer Science: characterization and early identification". In: *Scientometrics* 113.3, pp. 1645–1663.

Dridi, Amna et al. (2018). "k-NN Embedding Stability for word2vec Hyper-Parametrisation in Scientific Text". In: *Discovery Science - 21st International Conference, DS 2018, Limassol, Cyprus, October 29-31, 2018, Proceedings*, pp. 328–343.

— (2019a). "DeepHist: Towards a Deep Learning-based Computational History of Trends in the NIPS". In: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pp. 1–8.

— (Dec. 2019b). "Leap2Trend: A Temporal Word Embedding Approach for Instant Detection of Emerging Scientific Trends". In: *IEEE Access* 7, pp. 176414–176428. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2957440.

— (2021). "Scholarly data mining: A systematic review of its applications". In: *WIREs Data Mining and Knowledge Discovery* 11.2, e1395. DOI: https://doi.org/10.1002/widm.1395. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1395.

Dubossarsky, Haim et al. (July 2019). "Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 457–470.

Dunne, Cody et al. (Dec. 2012). "Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization". In: *Journal of the Association for Information Science & Technology* 63.12, pp. 2351–2369.

Effendy, Suhendry, Irvan Jahja, and Roland H.C. Yap (2014). "Relatedness Measures Between Conferences in Computer Science: A Preliminary Study Based on DBLP". In: *Proceedings of the 23rd International Conference on World Wide Web*. Seoul, Korea, pp. 1215–1220.

Effendy, Suhendry and H.C Yap Roland (2016). "The Problem of Categorizing Conferences in Computer Science". In: *Research and Advanced Technology for Digital Libraries*. Ed. by Norbert Fuhr et al. Cham: Springer International Publishing, pp. 447–450.

Effendy, Suhendry and Roland H.C. Yap (2017). "Analysing Trends in Computer Science Research: A Preliminary Study Using The Microsoft Academic Graph". In: *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1245–1250.

Erk, Katrin (2012). "Vector Space Models of Word Meaning and Phrase Meaning: A Survey." In: *Language and Linguistics Compass* 6.10, pp. 635–653.

Feng, Xia et al. (2017). "Big Scholarly Data: A Survey". In: *IEEE Trans. Big Data* 3.1, pp. 18–35.

Ferreira, Anderson A., Marcos André Gonçalves, and Alberto H.F. Laender (Aug. 2012). "A Brief Survey of Automatic Methods for Author Name Disambiguation". In: *SIGMOD Rec.* 41.2, pp. 15–26.

Firth, J.R. (1957). "A synopsis of linguistic theory 1930-55". In: 1952-59, pp. 1–32.

Fortunato, Santo et al. (2018). "Science of science". In: *Science* 359.6379.

Friedman, Carol et al. (June 2001). "GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles". In: *Bioinformatics* 17.suppl_1, S74–S82.

Furnas, George W. et al. (June 1988). "Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure". In: *SIGIR'88, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 465–480.

Gaber, Mohamed Medhat (2010). "Introduction". In: *Scientific Data Mining and Knowledge Discovery - Principles and Foundations*. Ed. by Mohamed Medhat Gaber. Springer, pp. 1–4. DOI: 10.1007/978-3-642-02788-8_1. URL: https://doi.org/10.1007/978-3-642-02788-8%5C_1.

Gleason, Henry A. (1961). *An introduction to descriptive linguistics*. Holt, Rinehart and Winston New York.

Godin, Benoît (July 2006). "On the origins of bibliometrics". In: *Scientometrics* 68.1, pp. 109–133.

Goldacre, B. (2008). *Bad Science*. HarperCollins Publishers. ISBN: 9780007283194. URL: https://books.google.tn/books?id=Gv1NQubrGNIC.

Grefenstette, Gregory (1994). *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

Gusenbauer, Michael (Nov. 2018). "Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases". In: *Scientometrics*. ISSN: 1588-2861.

Gusenbauer, Michael and Neal R. Haddaway (2020). "Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources". In: *Research Synthesis Methods* 11.2, pp. 181–217.

Hall, David, Daniel Jurafsky, and Christopher D. Manning (2008). "Studying the History of Ideas Using Topic Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '08. Association for Computational Linguistics, pp. 363–371.

Hamilton, William L., Jure Leskovec, and Dan Jurafsky (2016a). "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In: *ACL (1)*.

Hamilton, William L. et al. (2016b). "Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora". In: *Proc. of Conference on Empirical Methods in Natural Language Processing, EMNLP'2016*, pp. 595–605.

Harris, Zellig (1954). "Distributional structure". In: *Word* 10.23, pp. 146–162.

He, Jiangen and Chaomei Chen (2018). "Predictive Effects of Novelty Measured by Temporal Embeddings on the Growth of Scientific Literature". In: *CoRR* abs/1801.09121.

He, Kaiming et al. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385.

Heffernan, Kevin and Simone Teufel (Apr. 2018a). "Identifying problems and solutions in scientific text". In: *Scientometrics*.

— (Apr. 2018b). "Identifying problems and solutions in scientific text". In: *Scientometrics*.

Ho, Tin Kam (1995). "Random Decision Forests". In: *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. ICDAR '95. IEEE Computer Society, pp. 278–282.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hoonlor, Apirak, Boleslaw K. Szymanski, and Mohammed J. Zaki (Oct. 2013). "Trends in Computer Science Research". In: *Commun. ACM* 56.10, pp. 74–83.

Hou, Jianhua, Xiucai Yang, and Chaomei Chen (May 2018). "Emerging trends and new developments in information science: a document co-citation analysis (2009–2016)". In: *Scientometrics* 115.2, pp. 869–892.

Huang, Minlie, Qiao Qian, and Xiaoyan Zhu (June 2017). "Encoding Syntactic Knowledge in Neural Networks for Sentiment Classification". In: *ACM Trans. Inf. Syst.* 35.3, 26:1–26:27.

Hutter, Frank, Hoos Holger, and Leyton Brown Kevin (2014). "An Efficient Approach for Assessing Hyperparameter Importance". In: *31st Int. Conf. on Machine Learning*, pp. 754–762.

Ide, Nancy and Jean Véronis (1998). "Word sense disambiguation: The state of the art". In: *Computational Linguistics* 24, pp. 1–40.

Inderjeet, Mani (1999). *Advances in Automatic Text Summarization*. Ed. by Maybury Mark T. Cambridge, MA, USA: MIT Press. ISBN: 0262133598.

Jensen, Karen and Jean-Louis Binot (July 1987). "Disambiguating Prepositional Phrase Attachments by Using On-line Dictionary Definitions". In: *Comput. Linguist.* 13.3-4, pp. 251–260.

Jeong, Yoo Kyung, Min Song, and Ying Ding (2014). "Content-based author cocitation analysis". In: *Journal of Informetrics* 8.1, pp. 197–211.

Jin, P., R. Sui, and Y. Zhang (Dec. 2010). "A Knowledge Based Method for Chinese Word Sense Induction". In: *2010 Fourth International Conference on Genetic and Evolutionary Computing*, pp. 248–251.

Johnson, Mark (2009). "How the Statistical Revolution Changes (Computational) Linguistics". In: *Proceedings of the EACL 2009 Workshop on the Interaction Between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?* Association for Computational Linguistics, pp. 3–11.

Joulin, Armand et al. (2016). "Bag of Tricks for Efficient Text Classification". In: *CoRR* abs/1607.01759.

Jurafsky, Daniel and James H. Martin (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International. ISBN: 9780135041963.

Kaempf, Mirko et al. (Dec. 2015). "The Detection of Emerging Trends Using Wikipedia Traffic Data and Context Networks". In: *PloS one* 10, e0141892.

Kaisler, S. et al. (Jan. 2013). "Big Data: Issues and Challenges Moving Forward". In: *2013 46th Hawaii International Conference on System Sciences*, pp. 995–1004.

Kaji, Nobuhiro and Hayato Kobayashi (2017). "Incremental Skip-gram Model with Negative Sampling". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 363–371.

Karen, Sparck Jones (1988). "Document Retrieval Systems". In: ed. by Peter Willett. Taylor Graham Publishing. Chap. A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pp. 132–142.

Karypis, G. and V. Kumar (1999). "Chameleon: hierarchical clustering using dynamic modeling". In: *Computer* 32.8, pp. 68–75.

Khan, Samiya et al. (July 2017). "A survey on scholarly data: From big data perspective". In: *Information Processing & Management* 53, pp. 923–944.

Kim, Sung, Derek Hansen, and Richard Helps (Jan. 2018). "Computing research in the academy: insights from theses and dissertations". In: *Scientometrics* 114.1, pp. 135–158.

Kim, Yoon et al. (2014). "Temporal Analysis of Language through Neural Language Models". In: *LTCSS@ACL*, pp. 61–65.

Kong, X. et al. (2018). "VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation". In: *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1.

Koza, John R. et al. (1996). "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming". In: *Artificial Intelligence in Design '96*. Ed. by John S. Gero and Fay Sudweeks. Dordrecht: Springer Netherlands, pp. 151–170. ISBN: 978-94-009-0279-4. DOI: 10.1007/978-94-009-0279-4_9.

Krizhevsky, Alex, Sutskever Ilya, and Hinton Geoffrey E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pp. 1097–1105.

Kula, Sebastian et al. (2020). "Sentiment Analysis for Fake News Detection by Means of Neural Networks". In: *Computational Science – ICCS 2020*. Ed. by Valeria V. Krzhizhanovskaya et al. Cham: Springer International Publishing, pp. 653–666. ISBN: 978-3-030-50423-6.

Kulkarni, Vivek et al. (2015). "Statistically Significant Detection of Linguistic Change". In: *Proc. of the 24th International Conference on World Wide Web*, pp. 625–635.

Kutuzov, Andrey, Erik Velldal, and Lilja Óvrelid (2017). "Tracing armed conflicts with diachronic word embedding models". In: *Proc. of the Events and Stories in the News Workshop*, pp. 31–36.

Kutuzov, Andrey et al. (2018). "Diachronic word embeddings and semantic shifts: a survey". In: *Proc. of the 27th International Conference on Computational Linguistics*, pp. 1384–1397.

Landauer, Thomas K and Susan T. Dutnais (1997). "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge". In: *PSYCHOLOGICAL REVIEW* 104.2, pp. 211–240.

Landes, CShari, Claudia Leacock, and Christiane Fellbaum (1998). "Building Semantic Concordances". In: *WordNet: An Electronic Lexical Database, Chapter 8*. MITP, pp. 199–216.

Le, Quoc V. et al. (2012). "Building High-level Features Using Large Scale Unsupervised Learning". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML'12, pp. 507–514.

Lenci, Alessandro (2018). "Distributional Models of Word Meaning". In: *Annual Review of Linguistics* 4.1, pp. 151–171.

Levy, Omer and Yoav Goldberg (2014). "Dependency-Based Word Embeddings". In: *52nd Annual Meeting of the Association for Computational Linguistics*, pp. 302–308.

Li, C et al. (2019). "Mining Dynamics of Research Topics Based on the Combined LDA and WordNet". In: *IEEE Access* 7, pp. 6386–6399.

Li, Chenliang et al. (Aug. 2017). "Enhancing Topic Modeling for Short Texts with Auxiliary Word Embeddings". In: *ACM Trans. Inf. Syst.* 36.2, 11:1–11:30.

Li, Huajing et al. (Dec. 2006). "CiteSeerx: An architecture and Web service design for an academic document search engine". In: *Proceedings of the 15th International Conference on World Wide Web*, pp. 883–884.

Lieberman, E. et al. (Oct. 2007). "Quantifying the evolutionary dynamics of language". In: *Nature* 449.7163, pp. 713–716.

Light, Robert P., David E. Polley, and Katy Börner (2014). "Open data and open code for big science of science studies". In: *Scientometrics* 101.2, pp. 1535–1551.

Lin, Dekang (1997). "Using Syntactic Dependency As Local Context to Resolve Word Sense Ambiguity". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 64–71.

— (1998a). "An Information-Theoretic Definition of Similarity". In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. Morgan Kaufmann Publishers Inc., pp. 296–304.

— (1998b). "Automatic Retrieval and Clustering of Similar Words". In: *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*. Association for Computational Linguistics, pp. 768–774.

Liu, J. et al. (Mar. 2018). "A Survey of Scholarly Data Visualization". In: *IEEE Access* 6, pp. 19205–19221.

Liu, Jiaying et al. (2019). "Shifu2: A Network Representation Learning Based Model for Advisor-advisee Relationship Mining". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1.

Liu, Yu et al. (2015). "Science Navigation Map: An Interactive Data Mining Tool for Literature Analysis". In: *Proceedings of the 24th International Conference on World Wide Web*. Florence, Italy, pp. 591–596.

Lu, Wei et al. (Apr. 2018a). "Functional structure identification of scientific documents in computer science". In: *Scientometrics* 115.1, pp. 463–486.

Lu, Wei et al. (Apr. 2018b). "Functional structure identification of scientific documents in computer science". In: *Scientometrics* 115.1, pp. 463–486.

Lv, Peng Hui et al. (Aug. 2011). "Bibliometric trend analysis on global graphene research". In: *Scientometrics* 88.2, pp. 399–419.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9, pp. 2579–2605.

Maflahi, Nabeil and Mike Thelwall (2018). "How quickly do publications get read? The evolution of mendeley reader counts for new articles". In: *Journal of the Association for Information Science and Technology* 69.1, pp. 158–167.

Martínez-Gómez, Aída (Apr. 2015). "Bibliometrics as a tool to map uncharted territory: A study on non-professional interpreting". In: 23, pp. 1–18.

McBurney, K. Melissa and L. Pamela Novak (Feb. 2002). "What is bibliometrics and why should you care?" In: *IEEE International Professional Communication Conference*, pp. 108–114.

McCulloch, Warren S and Walter Pitts (1988). "Neurocomputing: Foundations of Research". In: ed. by James Anderson and Edward Rosenfeld. MIT Press. Chap. A Logical Calculus of the Ideas Immanent in Nervous Activity, pp. 15–27.

Mei, Qiaozhu and ChengXiang Zhai (Jan. 2008). "Generating Impact-Based Summaries for Scientific Literature." In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pp. 816–824.

Merediz, Sola Ignasi and Bariviera F. Aurelio (2019). "A bibliometric analysis of bitcoin scientific production". In: *Research in International Business and Finance* 50, pp. 294–305.

Mikolov, Tomas, Yih Wen-Tau, and Zweig Geoffrey (2013a). "Linguistic Regularities in Continuous Space Word Representations." In: *HLT-NAACL*, pp. 746–751.

— (2013b). "Linguistic Regularities in Continuous Space Word Representations." In: *NAACL-HLT*, pp. 746–751.

Mikolov, Tomas et al. (2013c). "Distributed Representations of Words and Phrases and Their Compositionality". In: *26th Int. Conf. on Neural Information Processing Systems*, pp. 3111–3119.

Mikolov, Tomas et al. (2013d). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* Pp. 3111–3119.

Mikolov, Tomas et al. (2013e). "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781.

Miller, George A. (Nov. 1995). "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11, pp. 39–41.

Miñarro-Giménez, José Antonio, Oscar Marín-Alonso, and Matthias Samwald (2015). "Applying deep learning techniques on medical corpora from the World Wide Web: a prototypical system and evaluation". In: *CoRR* abs/1502.03682.

Mingers, John and Loet Leydesdorff (2015). "A review of theory and practice in scientometrics". In: *European Journal of Operational Research* 246.1, pp. 1–19.

Mitchell, Tom (1997). *Machine Learning*. McGraw-Hill Education.

Monroy, Sonia E. and Hernando Diaz (Apr. 2018). "Time series-based bibliometric analysis of the dynamics of scientific production". In: *Scientometrics*.

Morin, F. and Y. Bengio (2005). "Hierarchical probabilistic neural network language model". In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 246–252.

Mortenson, Michael J. and Richard Vidgen (2016). "A Computational Literature Review of the Technology Acceptance Model". In: *Int. J. Inf. Manag.* 36.6, pp. 1248–1259.

Mumtazimah, Mohamad, Lazim Yuzarimi, and Rosle Suharmili (July 2018). "Academic Social Network Sites: Opportunities and Challenges". In: *International Journal of Engineering & Technology* 7, p. 133.

Nabout, João Carlos et al. (May 2018). "Do traditional scientometric indicators predict social media activity on scientific knowledge? An analysis of the ecological literature". In: *Scientometrics* 115.2, pp. 1007–1015.

Nicolai Meinshausen and Peter Buhlmann (2010). "Stability selection". In: *Royal Statistical Society* 72.4, pp. 417–473.

Nuzzolese, Andrea Giovanni et al. (2016). "Conference Linked Data: The ScholarlyData Project". In: *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference*, pp. 150–158.

Olaf, Ronneberger, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597.

Orkphol, Korawit and Wu Yang (2019). "Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet". In: *Future Internet* 11, p. 114.

Osborne, Francesco and Enrico Motta (2012). "Mining Semantic Relations between Research Areas". In: *The Semantic Web – ISWC 2012*. Ed. by Cudré Mauroux Philippe et al. Springer Berlin Heidelberg, pp. 410–426.

Osborne, Francesco, Enrico Motta, and Paul Mulholland (2013). "Exploring Scholarly Data with Rexplore". In: *The Semantic Web – ISWC 2013*. Ed. by Alani Harith et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 460–477.

Osgood, Charles E., Murray S. Miron, and William H. May (1975). *Cross-cultural universals of affective meaning [by] Charles E. Osgood, William H. May [and] Murray S. Miron*. English. University of Illinois Press Urbana, xix, 486 p.

Pado, Sebastian and Mirella Lapata (2007). "Dependency-based Construction of Semantic Space Models". In: *Computational Linquistics* 33.2, pp. 161–199.

Paul, Michael and Roxana Girju (2009). "Topic modeling of research fields: An interdisciplinary perspective". English (US). In: *International Conference Recent Advances in Natural Language Processing, RANLP*, pp. 337–342. ISSN: 1313-8502.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014a). "Glove: Global Vectors for Word Representation". In: *EMNLP*. Vol. 14, pp. 1532–1543.

— (2014b). "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14, pp. 1532–1543.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith (2019). "To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks". In: *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*. Ed. by Isabelle Augenstein et al. Association for Computational Linguistics, pp. 7–14. DOI: 10.18653/v1/w19-4302.

Piatetsky-Shapiro, G. (1991). "Discovery, analysis and presentation of strong rules". In: *Knowledge Discovery in Databases*. AAAI Press, pp. 229–248.

Pilkington, Adrian (2004). "Defining technology management: a citation/co-citation study". In: *2004 IEEE International Engineering Management Conference (IEEE Cat. No.04CH37574)* 1, 337–341 Vol.1.

Polikar, R. (2006). "Ensemble based systems in decision making". In: *IEEE Circuits and Systems Magazine* 6.3, pp. 21–45.

Priem, Jason and Kaitlin Light Costello (2010). "How and Why Scholars Cite on Twitter". In: *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem*. Vol. 47. American Society for Information Science, 75:1–75:4.

Qazvinian, Vahed and Dragomir R. Radev (2008). "Scientific Paper Summarization Using Citation Summary Networks". In: *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*. COLING '08. Association for Computational Linguistics, pp. 689–696.

Quinlan, J. R. (1990). "Machine Learning". In: ed. by Yves Kodratoff and Ryszard S. Michalski. Morgan Kaufmann Publishers Inc. Chap. Probabilistic Decision Trees, pp. 140–152.

Rada, Mihalcea and Nastase Vivi (2012). "Word Epoch Disambiguation: Finding How Words Change Over Time". In: *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 259–263.

Rahul, Jha, Abu Jbara Amjad, and Radev Dragomir (Aug. 2013). "A System for Summarizing Scientific Topics Starting from Keywords". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 572–577.

Reed, J. W. et al. (Dec. 2006). "TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams". In: *2006 5th International Conference on Machine Learning and Applications*, pp. 258–263.

Rexha, Andi et al. (Apr. 2018). "Authorship identification of documents with high content similarity". In: *Scientometrics* 115.1, pp. 223–237.

Rinaldo, Alessandro et al. (2012). "Stability of Density-based Clustering". In: *Journal of Machine Learning Research* 13.1, pp. 905–948.

Robertson, Stephen E. et al. (Nov. 1992). "Okapi at TREC". In: *Proceedings of The First Text REtrieval Conference, TREC*, pp. 21–30.

Romijn, Jan-Willem (2014). *Philosophy of statistics*.

Rosin, Guy D., Eytan Adar, and Kira Radinsky (2017). "Learning Word Relatedness over Time". In: *Proc.of Conference on Empirical Methods in Natural Language Processing, EMNLP'2017*, pp. 1168–1178.

Rossetto, Dennys Eduardo et al. (Mar. 2018). "Structure and evolution of innovation research in the last 60 years: review and future trends in the field of business through the citations and co-citations analysis". In: *Scientometrics*.

Safder, Iqra and Saeed-Ul Hassan (Sept. 2018). "DS4A: Deep Search System for Algorithms from Full-text Scholarly Big Data". In: DOI: 10.1109/ICDMW.2018.00186.

Sahel, José Alain (2011). "Quality Versus Quantity: Assessing Individual Research Performance". In: *Science Translational Medicine* 3.84, pp. 84cm13–84cm13.

Salatino, Angelo Antonio, Francesco Osborne, and Enrico Motta (2017). "How are topics born? Understanding the research dynamics preceding the emergence of new areas". In: *PeerJ Computer Science* 3, e119.

— (2018). "AUGUR: Forecasting the Emergence of New Research Topics". In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*, pp. 303–312.

Salton, G., A. Wong, and C.S. Yang (Nov. 1975). "A Vector Space Model for Automatic Indexing". In: *Commun. ACM* 18.11, pp. 613–620.

Salton, Gerard (1968). *Automatic Information Organization and Retrieval*. McGraw Hill Text. ISBN: 0070544859.

Salton, Gerard and Michael J. McGill (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. ISBN: 0070544840.

Sebastiani, Fabrizio (Mar. 2002). "Machine Learning in Automated Text Categorization". In: *ACM Comput. Surv.* 34.1, pp. 1–47.

Shadish, R. William et al. (Aug. 1995). "Author Judgements about Works They Cite: Three Studies from Psychology Journals". In: 25, pp. 477–498.

Shaoul, Cyrus and Chris Westbury (May 2010). "Exploring lexical co-occurrence space using HiDEx". In: *Behavior Research Methods* 42.2, pp. 393–413.

Shardlow, Matthew et al. (2018). "Identification of research hypotheses and new knowledge from scientific literature". In: *BMC Medical Informatics and Decision Making* 18.1.

Shi, Lei et al. (Dec. 2015). "VEGAS: Visual influEnce GrAph Summarization on Citation Networks". In: *IEEE Transactions on Knowledge and Data Engineering* 27.12, pp. 3417–3431.

Singh, Harsimran and Vishal Gupta (May 2015). "An Insight into Word Sense Disambiguation Techniques". In: *International Journal of Computer Applications* 118.23, pp. 32–39.

Singhal, Amit, Chris Buckley, and Mandar Mitra (1996). "Pivoted Document Length Normalization". In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 21–29.

Soriano, Alba Santa, Carolina Lorenzo Álvarez, and Rosa María Torres Valdés (May 2018). "Bibliometric analysis to identify an emerging research area: Public Relations Intelligence—a challenge to strengthen technological observatories in the network society". In: *Scientometrics*.

Sproat, Richard, Julia Hirschberg, and David Yarowsky (1992). "A corpus-based synthesizer". In: *Proceedings of the International Conference on Spoken Language Processing*.

Steyvers, Mark and Tom Griffiths (2007). "Probabilistic Topic Models". In: *Handbook of Latent Semantic Analysis*. Ed. by T. Landauer et al. Lawrence Erlbaum Associates.

Stone, P.J. (1969). "Improved quality of Content-Analysis Categories: Computerized-Disambiguation Rules for High-Frequency English Words". In: *In Gerbner, George; Holsti, Ole, Krippendorf, Klaus; Paisley, William J.; and Stone, Philip J. (Eds.), The Analysis of Communication Content, John Wiley and Sons,* pp. 199–233.

Stone, Philip J. and Earl B. Hunt (1963). "A Computer Approach to Content Analysis: Studies Using the General Inquirer System". In: *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*. ACM, pp. 241–256.

Sun, Xiaoling et al. (2011). "Detecting Ambiguous Author Names in Crowdsourced Scholarly Data". In: *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pp. 568–571.

Swain, MC and JM Cole (Oct. 2016). "ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature". In: *Journal of Chemical Information and Modeling*, pp. 1894–1904.

Szymanski, Terrence (2017). "Temporal Word Analogies: Identifying Lexical Replacement with Diachronic Word Embeddings". In: *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 448–453.

Taigman, Yaniv et al. (2014). "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708.

Tan, Zhaowei et al. (2016). "AceMap: A Novel Approach Towards Displaying Relationship Among Academic Literatures". In: *Proceedings of the 25th International Conference Companion on World Wide Web*. WWW '16 Companion, pp. 437–442.

Tang, Jie (2016). "AMiner: Toward Understanding Big Scholar Data". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. WSDM '16, pp. 467–467.

Tang, Jie et al. (2008). "ArnetMiner: Extraction and Mining of Academic Social Networks". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 990–998.

Tao, Shibo et al. (2017). "From Citation Network to Study Map: A Novel Model to Reorganize Academic Literatures". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion, pp. 1225–1232.

Thelwall, Mike (2018). "Differences between journals and years in the proportions of students, researchers and faculty registering Mendeley articles". In: *Scientometrics* 115, pp. 717–729.

Trujillo, Caleb M. and Tammy M. Long (2018). "Document co-citation analysis to enhance transdisciplinary research". In: *Science Advances* 4.1.

Tuarob, Suppawong et al. (2016). "AlgorithmSeer: A System for Extracting and Searching for Algorithms in Scholarly Big Data". In: *IEEE Trans. Big Data* 2.1, pp. 3–17.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Behavioral Science: Quantitative Methods. Reading, Mass.: Addison-Wesley.

Turney, Peter D. and Patrick Pantel (2010). "From Frequency to Meaning: Vector Space Models of Semantics". In: *J. Artif. Int. Res.* 37.1, pp. 141–188.

Vahe, Tshitoyan et al. (July 2019). "Unsupervised word embeddings capture latent knowledge from materials science literature". In: *Nature* 571.7763, pp. 95–98.

Véronis, Jean (2004). "HyperLex: lexical cartography for information retrieval". In: *Computer Speech & Language* 18, pp. 223–252.

Virginia, Cano (July 1989). "Citation behavior: Classification, utility, and location". In: 40, pp. 284–290.

Wang, Bin et al. (Feb. 2019). "Evaluating Word Embedding Models: Methods and Experimental Results". In: *CoRR* abs/1901.09785.

Weber, Rosina and S. Gunawardena (Jan. 2011). "Representing scientific knowledge". In: pp. 279–283.

Weismayer, Christian and Ilona Pezenka (Dec. 2017). "Identifying emerging research fields: a longitudinal latent semantic keyword analysis". In: *Scientometrics* 113.3, pp. 1757–1785.

Weller, Katrin, Evelyn Droge, and Cornelius Puschmann (2011). "Citation analysis in Twitter. In: Approaches for Defining and Measuring Information Flows within Tweets during Scientific Conferences". In: *sharp MSM2011, 1st Workshop on Making Sense of Microposts*, pp. 1–12.

Wittgenstein, Ludwig (1953). *Philosophical Investigations*. Oxford: Basil Blackwell.

Word2Vec Tutorial Part I: The SkipGram Model (n.d.). http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf. Accessed: 2019-03-29.

Word2Vec Tutorial Part II: The Continuous Bag-of-Words Model (n.d.). http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_II_The_Continuous_Bag-of-Words_Model.pdf. Accessed: 2019-03-12.

Wu, Z. et al. (2014). "Towards building a scholarly big data platform: Challenges, lessons and opportunities". In: *IEEE/ACM Joint Conference on Digital Libraries*, pp. 117–126.

Yao, Zijun et al. (2018a). "Dynamic Word Embeddings for Evolving Semantic Discovery". In: *Proc. the 11th ACM International Conference on Web Search and Data Mining*, pp. 673–681.

— (2018b). "Dynamic Word Embeddings for Evolving Semantic Discovery". In: *11th ACM Int. Conf. on Web Search and Data Mining*, pp. 673–681.

Yarowsky, David (1992). "Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora". In: *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*. COLING '92. Association for Computational Linguistics, pp. 454–460.

— (1994). "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French". In: *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*. ACL '94. Las Cruces, New Mexico: Association for Computational Linguistics, pp. 88–95.

— (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods". In: *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 189–196.

Yehoshua, Bar-Hillel (1960). "The Present Status of Automatic Translation of Languages". In: *Advances in Computers* 1, pp. 91–163.

Ying, Ding et al. (Sept. 2014). "Content-based Citation Analysis: The Next Generation of Citation Analysis". In: *J. Assoc. Inf. Sci. Technol.* 65.9, pp. 1820–1833.

Zehra, Taskin and Al Umut (2018). "A content-based citation analysis study based on text categorization". In: *Scientometrics* 114.1, pp. 335–357.

Zhang, Chao and Jiancheng Guan (Nov. 2017). "How to identify metaknowledge trends and features in a certain research field? Evidences from innovation and entrepreneurial ecosystem". In: *Scientometrics* 113.2, pp. 1177–1197.

Zhang, Daokun et al. (2018). "Network Representation Learning: A Survey". In: *CoRR* abs/1801.05852.

Zhang, J. et al. (2018). "PePSI: Personalized Prediction of Scholars' Impact in Heterogeneous Temporal Academic Networks". In: *IEEE Access* 6, pp. 55661–55672.

Zhang, Y. et al. (Oct. 2016). "The Past is Not a Foreign Country: Detecting Semantically Similar Terms across Time". In: *IEEE Transactions on Knowledge and Data Engineering* 28.10, pp. 2793–2807.

Zhao, Shu et al. (Mar. 2018a). "A novel classification method for paper-reviewer recommendation". In: *Scientometrics*, pp. 1–21.

Zhao, Shu et al. (Mar. 2018b). "A novel classification method for paper-reviewer recommendation". In: *Scientometrics*, pp. 1–21.

Zhao, Star and Fred Ye (July 2013). "Power-law link strength distribution in paper cocitation networks". In: *Journal of the American Society for Information Science and Technology* 64, pp. 1480–1489.