PREDICTING THE ACADEMIC INFLUENCE AND TRENDING RESEARCH TOPICS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT YÜKSELEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

SEPTEMBER 2022

Approval of the thesis:

PREDICTING THE ACADEMIC INFLUENCE AND TRENDING RESEARCH TOPICS

submitted by **MURAT YÜKSELEN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering, METU** _____

Assist. Prof. Dr. Alev Mutlu
Co-supervisor, **Computer Engineering, Kocaeli University** _____

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Prof. Dr. Osman Abul
Computer Engineering, TOBB ETU _____

Assoc. Prof. Dr. Pelin Angın
Computer Engineering, METU _____

Assist. Prof. Dr. Burcu Yılmaz
Institute of Information Technologies, Gebze Technical University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Murat Yükselen

Signature :

ABSTRACT

PREDICTING THE ACADEMIC INFLUENCE AND TRENDING RESEARCH TOPICS

Yükselen, Murat
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Pınar Karagöz
Co-Supervisor: Assist. Prof. Dr. Alev Mutlu

September 2022, 88 pages

Predictions on academic research are thoroughly studied in the literature. In this thesis, we focus on two prediction problems in this domain. First we study the problem of topic adoption prediction for an author within a social academic network. We model the problem with an influence detection point of view, and propose that the influence on the author is an important factor. Hence, we define a novel influencee prediction based feature and developed an algorithm to calculate the influence propagated towards the author. The effect of this feature is explored together with and in comparison to other features used in the literature for the problem. The experiments conducted on Arnet Miner data set show that accumulated influence on author is effective for predicting topic adoption.

As a second problem, we try to enlarge our scope and generalize by focusing on predicting the trending research topics from a collection of academic papers. Previous efforts model the problem in different ways and mostly apply classical approaches such as correlation analysis and clustering. There are also several recent neural model based solutions, however they rely on feature vectors and additional

information for the trend prediction. In this work, given a collection of publications within the observation time window, we predict whether the use of a keyword will increase, decrease or be steady for the future time window (prediction window). As the solution, we propose a family of deep neural architectures that focus on generating summary representations for paper collections under the query keyword. Due to the sequence based nature of the data, Long Short-Term Memory (LSTM) module plays a core role, but it is combined with different layers in a novel way. The first group of proposed neural architectures consider each paper as a sequence of keywords and use word embeddings to construct paper collection representations. In this group, the proposed architectures differ from each other in the way year based and overall summary representations are constructed. In the second group, each paper is directly represented as a vector and the use of different paper embedding techniques are explored. The analyses of the models are performed on a variety of paper collections belonging to different academic venues, obtained from Microsoft Academic Graph data set. The experiments conducted against baseline methods show that proposed deep neural based models achieve higher trend prediction performance than the baseline models on the overall. Among the proposed models, paper embedding based models provide better results for most of the cases.

Keywords: Information flow, Link prediction, Trend prediction, Keyword popularity prediction, Deep learning, Document vector, Classification, Social networks

ÖZ

AKADEMİK ETKİYİ VE ARAŞTIRMA KONULARININ EĞİLİMİNİ TAHMİNLEME

Yükselen, Murat
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Pınar Karagöz
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Alev Mutlu

Eylül 2022 , 88 sayfa

Akademik çalışmalara ilişkin tahminleme literatürde kapsamlı bir şekilde incelenmektedir. Bu tezde bu alandaki iki tahmin problemine odaklanılmaktadır. İlk olarak, bir sosyal akademik ağ içindeki bir yazar için konu benimseme tahminleme problemi incelenmektedir. Sorun bir etki tespit bakış açısıyla modellenip ve yazar üzerindeki etkinin önemli bir faktör olduğu önerilmektedir. Bu nedenle etki alımına dayalı yeni bir özellik tanımlandı ve yazara yayılan etkiyi hesaplamak için bir algoritma geliştirildi. Bu özelliğin etkisi, problem için literatürde kullanılan diğer özelliklerle birlikte ve bunlarla karşılaştırılarak araştırıldı. Arnet Miner veri seti üzerinde yapılan deneyler, yazar üzerinde birikmiş etkinin, konunun benimsenmesini öngörmede etkili olduğunu göstermektedir.

İkinci problem olarak, bir akademik makale koleksiyonundan araştırma konularının eğilimini tahmin etmeye odaklanarak kapsamımızı genişletmeye ve genelleştirmeye çalışıyoruz. Önceki çalışmalar sorunu farklı şekillerde modellemekte ve çoğunlukla korelasyon analizi ve kümeleme gibi klasik yaklaşımları uygulamaktadır. Ayrıca bir-

kaç yeni sinir modeli tabanlı çalışma vardır, ancak bunlar eğilim tahmini için özellik vektörlerine ve ek bilgilere ihtiyaç duyarlar. Bu çalışmada gözlem zaman dilimi içinde bir yayın koleksiyonu verildiğinde, sonraki zaman dilimi (tahmin zaman dilimi) için sorgulanan bir anahtar kelime kullanımının artacağı, azalacağı veya sabit kalacağı tahminleniyor. Çözüm olarak sorgu anahtar kelimesi için yayın koleksiyonlarından özet temsiller oluşturmaya odaklanan bir takım derin nöral mimarileri öneriyoruz. Verilerin sıra tabanlı yapısı nedeniyle Uzun Kısa-Süreli Bellek (LSTM) modülü temel bir rol oynar, ancak farklı katmanlarla yeni bir şekilde birleştirilir. Önerilen nöral mimarilerin ilk grubu, her makaleyi bir anahtar sözcük dizisi olarak kabul eder ve yayın koleksiyonu temsillerini oluşturmak için kelime temsilinden başlanır. Bu grupta önerilen mimariler yıl bazlı ve genel özet temsillerin oluşturulma şekli bakımından birbirinden farklıdır. İkinci grupta her yayın doğrudan bir vektör olarak temsil edilir ve farklı doküman temsil tekniklerinin kullanımı araştırılır. Modellerin analizleri Microsoft Academic Graph veri setinden elde edilen farklı akademik mekanlara ait çeşitli yayın koleksiyonları üzerinde gerçekleştirilmiştir. Temel yöntemlere karşı yapılan deneyler, önerilen derin sinir tabanlı modellerin genel olarak temel modellerden daha yüksek eğilim tahmin performansı elde ettiğini göstermektedir. Önerilen modeller arasında yayın temsili tabanlı modeller çoğu durumda daha iyi sonuçlar vermektedir.

Anahtar Kelimeler: Bilgi akışı, Bağlantı tahminleme, Eğilim tahminleme, Anahtar kelime popülerlik tahminleme, Derin öğrenme, Doküman vektörü, Sınıflandırma, Sosyal ağlar

To my dear wife Merve and lovely son Dağhan Ata

ACKNOWLEDGMENTS

This study was a very long journey from its start in 2010 into 2022. This long process was sometimes slow and ambiguous and sometimes gained momentum and fruitful. I would like to express my deepest gratitude to everyone who has been a part of this journey. Doing a Ph.D. while working full-time was not an easy experience.

First of all, I would like to express my deepest gratitude to my thesis supervisors Prof.Dr. Pınar Karagöz and my co-supervisor Assist. Prof. Dr. Alev Mutlu. Their guidance, broad knowledge and supportive attitude have always been of great value during this research. During this long journey they could have helped to finish more students and I really feel lucky that I was able to study with them.

I would like to thank Prof. Dr. İsmail Hakkı Toroslu and Prof. Dr. Osman Abul for their valuable suggestions and comments throughout the steering meetings of this study. I would also like to thank Assoc. Prof. Dr. Pelin Angın and Assist. Prof. Dr. Burcu Yılmaz for kindly accepting being in the examining committee.

I would like to thank my employers Environmental Tectonics Corporation and Somera A.Ş. for providing unconditional leaves to pursue my studies and to my supportive co-workers.

I would like to acknowledge that this work initially supported by TÜBİTAK (Scientific and Technical Research Council of Turkey) 2211 Scholarship program. Also this work has been partially supported by TÜBİTAK with grant number 117E566. I am grateful to receive these supports during my studies.

This long journey also witnessed the inception of my own family too. Merve knew from the start that I have been pursuing Ph.D. studies but I am sure she was not expecting this to be this long. From the early days of our marriage to the fruitful and intense childrearing days, she was always there supporting me. Pursuing a Ph.D. while working full-time is very exhausting and expects devotion. There are countless

times I had to sacrifice quality time with my family and put into my Ph.D. studies to advance. Sometimes I had to excuse myself from family trips where I still could not have a chance to go with any of them again. Although it is not enough in turn, I dedicate this work to my wife Merve and son Dağhan Ata. Dağhan Ata Yükselen, hope these lines encourage you to pursue your dreams whenever you read, now it is time for us to play.

Finally, my parents and brother whose encouragement and support was always there without any limits. It is very hard to write into words but I am very lucky to get all this love, unconditional understanding and support. I am very happy to have you all in my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
Abbr	Abbreviation
AMiner	Arnet Miner
Avg	Average
CNN	Convolutional Neural Network
LinReg	Linear Regression
LogReg	Logistic Regression
LSTM	Long Short-Term Memory
MAG	Microsoft Academic Graph
NaN	Not a Number
RNN	Recurrent Neural Network
ROC	Receiving Operational Characteristic
SVC	Support Vector Classification
SVM	Support Vector Machines
SVR	Support Vector Regression

CHAPTER 1

INTRODUCTION

Scientific topics evolve continuously and predicting their popularity and adoption are valuable tools that perform analysis on community and individual levels respectively. Predicting trends has always been an attractive ability in different domains. From fashion to social media, analyzing the current condition and predicting the future behavior brings advantages such as being the first or early adopter of the trend. Furthermore, trend prediction can help policymakers to implement necessary actions. Numerous research to predict trends has been conducted on several domains such as financial markets [1, 2], public health issues [3], and environmental issues [4].

Social network analysis is concerned with analysing the structure of the network and behaviour of individuals forming the network [5]. Although early studies in social network analysis focused on building descriptive models, with increasing amount of social network data, the research direction moved to building predictive models [6]. Such predictive models can be used in a variety of domains such as link prediction for friend recommendation [7, 8], influence detection for advertisement domain [9], and community detection for urban safety domain [10].

In this study, we work on *academic social networks* (also called *scientific collaboration networks* [11]) with a predictive point of view. In academic social networks, nodes represent entities such as authors and papers and edges represent relations such as authoring a paper and co-authorship. There are various studies in the literature on academic social networks aiming to predict collaboration patterns [12, 13, 14].

In academic research, data analytics and prediction techniques have been used in a variety of different problems such as publication prediction, collaboration analysis

and academic team formation [15, 16, 17]. As another important problem, predicting the trend of the topics has several benefits for academic research community. Such insights may help funding agencies to optimize their policies [18] and guide technology companies to shape their policies [19]. In addition, knowing future research trends may help new researchers to plan their studies [20].

With the recent advancements in machine learning, deep learning based solutions have been devised for prediction problems in variety of domains [21], also successful results for various text mining and information extraction problems using recent deep neural models have been published. Academic research topic trend prediction problem has been studied earlier, but mostly through more conventional data mining and machine learning approaches [22, 19]. There are recent efforts focusing on research topic prediction [23, 24, 25] and topic trend prediction [26], but they rely on hand-crafted features and additional information such as a semantic network, citation information or influence among research fields and venues. There are also related studies in the literature to detect hot topics [27] rather than trend prediction.

1.1 Motivation and Problem Definition

We study topic prediction both at the individual and the community level.

For individual level prediction, apart from the studies in the literature on academic social networks aiming to predict collaboration patterns [12, 13, 14], we focus on the problem of topic adoption, and propose a method to predict topic adoption of an author. More specifically, given an author and a new topic for the author, we aim to predict whether the author will publish a work on the given topic in the next time frame.

At the community level, we approached the problem as topic trend prediction and focused on popularity changes. In this work, we aim to explore whether deep neural architectures can be more effective in coping with topic trend prediction problem without using hand-crafted features and external information. To this aim, we propose a set of novel neural models that use only the paper collections for processing. In the first three architectures we focus on employing word representations within

different neural architectures. In the next three proposed models, we explore the use of different paper representations within the proposed neural architecture.

We formulate the challenged problem, trend detection of academic research topics, as predicting the trend of the keywords that describe topics. More specifically, we assume that a research topic can be described as a set of keywords, and we aim to predict the trend of a keyword. This assumption has been also used in related studies such as [27], [28], and also topic modeling studies in general. We model the trend prediction problem as a *supervised learning* problem with three labels such that given a keyword, we aim to predict as to whether its use will *increase*, *decrease* or will stay *steady*.

1.2 Proposed Methods and Models

Two main problems are attacked in this dissertation can be listed as topic adoption prediction and topic popularity prediction.

For topic adoption prediction problem, we argue that the amount of influence accumulated on the author for the given topic can affect the topic adoption. To this aim, we propose a new feature, *influencee score*, and propose an algorithm to compute this feature. The proposed algorithm is inspired from the influence detection method given in [29], in which the amount of influence going out from an author to other authors within social stream is computed. The effect of this new feature for topic adoption prediction is analyzed within a multiple logistic regression model together with and in comparison to the features given in [11]. Additionally, we compare the performance against the baseline model of [11], as well.

In topic popularity prediction problem, our second focus, we modeled it as both regression and classification problem where we try to predict any given keyword popularity trend will increase, steady, or decline.

One can consider various ways to set the labels for increasing, decreasing, and steady use of keywords. We define this behavior of trend in terms of frequency distributions. For a given time window, the label is determined based on the past observation of the

frequency distribution of the keyword. More specifically, we can informally define the keyword trend prediction problem as follows: Given a sequence of published papers in temporal order for a venue and a query keyword, the aim is to predict the trend label for the query keyword for the future time window.

The proposed neural architectures base on generating summary representations of the observed publications (in the observation window) in order to generate trend prediction of the query keyword (for the prediction window). Therefore input to the models is paper collections and a query keyword, and output is a trend label prediction. Since the textual data is represented as a sequence of tokens, Long Short-Term Memory (LSTM) neural model is used as a core component of the architectures. However it is combined with other modules (including other LSTM modules) in a novel setting for the focused prediction problem.

We propose two groups of architectures on the basis of using word embedding or paper embedding in the processing. Within the first group, we propose three architectures, exploring the use of only year based summary (in Model 1), the use of both year based and observation dimension (in Model 2), and the use of observation dimension where year based summary is constructed by a convolution layer (Model 3). In the second group of neural architectures, we use year based and observation dimension summary representations, but this time, explore the use of different paper embedding modules, LSTM based paper embedder module (Model 4), doc2vec [30] (Model 5) and Specter [31] (Model 6).

We analyze the performance of the proposed methods on a collection of academic papers from several well-known conferences along a timeline of 13 years. The analysis is conducted per venue for a collection of test query keywords. The selected conferences have overlapping focus, but also each has its own theme and academic community. Therefore, by venue based analysis we aim to predict the trend within each theme and community. Additionally, we conduct trend prediction analysis by combining the paper collection of all the venues. With this, we additionally explored the prediction performance under a higher volume of publications and more evidence for the trend to gain insight about the trend in a broader research field.

1.3 Contributions and Novelties

Our contributions are as follows:

- Explored influence in social networks by focusing on the receiving end with *Influencee Score* feature for topic propagation.
- An algorithm to calculate *Influencee Score* feature.
- Analysis of the proposed *Influencee Score* feature with respect to other compatible features.
- Formal definition of the research topic trend prediction is built on top of keyword popularity prediction for a history of published papers collection.
- Proposed keyword and paper level deep neural network models in topic popularity prediction. Each model explores alternative representations of paper summaries.
- Experiments to analyze the performance of models with respect to state of the art document based models and baseline models.
- A qualitative analysis is given to explore the performance of the proposed models.

1.4 The Outline of the Thesis

In Chapter 2 an overview of the related studies are presented. In Chapter 3 information flow in social networks are discussed and influencee score feature is examined. In Chapter 4 topic popularity prediction is explored and deep neural models are evaluated at consuming the academic network content on keyword level and document level. Finally, Chapter 5 discusses the findings of proposed techniques and evaluation results with future directions.

CHAPTER 2

RELATED WORK

In this chapter, related studies in the literature are given and their relations to our works appearing in the following chapters are given.

In Section 2.1, we summarize the related studies in the literature on influence detection and social network analysis on scientific collaboration networks. Then studies on academic trend prediction is summarized in Section 2.2. Finally in Section 2.3 the document embedding related studies are briefly described.

2.1 Social networks and influence detection

In a social network, action correlations of the agents are studied and defined as a result of social influence, homophily or confounding (environment) effects [32]. In [33], homophily and influence are studied within various sociological aspects, from relationship types to different sized communities. Network of ties, connections between individuals are affected heavily from homophily and in turn open to receive more influence. Combined effects of social influence and homophily are studied in large scale networks such Wikipedia and LiveJournal in [34]. Social influence and homophily effects are investigated through randomization tests on first-order effects, leaving second-order effects such as community and structural similarity as a future work [35]. The problem of distinguishing social influence and homophily is studied in [36]. In [32], social influence is studied to be more identifiable among other correlations.

In [12], collaboration patterns of authors in scientific papers are studied through sta-

tistical network features. It is reported that coauthor networks tend to include simple collaboration patterns. The study in [13] extends the scientific collaboration network structure with citation information in order to analyze topic modeling and evolution. In [14], the authors further use the text content as well as the network structure. They aim to track the popularity of the events and discover the evolution of the events over time as event diffusion on Twitter and DBLP. In [37], individual collaboration networks are studied in order to predict the evolution of collaboration. The study focuses on social collaboration network under computer science field on a 25 year time-window both at community level and individual level. In [29], the authors analyze scientific collaboration network in order to predict topic adoption. The topic similarity and co-author topic adoption are reported as features effective on topic adoption prediction.

Information diffusion [38] is studied on various types of social networks such as blogspace [39]. Through user behavior modeling on features such as neighborhood, topic and recipient, in [40], communication flow predictability is studied on MySpace network. In [41], the authors studied information diffusion on blogosphere data without incorporating post contents. It is reported that information cascades mainly as a tree. Feature implementations of the this work is also compatible with this observation. In [42], retweeting on Twitter is studied as an information diffusion problem for behavior modeling. The authors use conditional random fields with features such as content influence, network influence and temporal decay. Within scientific collaboration network including author-topic interaction, the work in [43] studies group and community growth and evolution. In [11], the authors aim to detect most influential authors in the academic social network and propose a social stream based solution.

Our work in Chapter 3 shares several common properties with previous studies on academic social networks. In addition to co-authorship network, we use textual content, as in [14], in limited to paper title and abstract. The most similar one among such studies is the one in [11], challenging the same problem of topic adoption prediction. However, the main difference of our work is that a novel feature is proposed. In order to realize this new feature, we incorporate ideas from the literature on information diffusion and influence detection. More specifically, the proposed algorithm for computing our new feature has its roots from the solution given in [29], however

the structure of the social stream is changed, and hence the proposed algorithm has considerable differences. Since we have adopted several basics from the works in [11] and [29], in Section 3.2 , we present further details for these two studies.

2.2 Studies on Academic Trend Prediction

Prediction tasks within the domain of academic research efforts have been an attractive problem to study. Citation recommendation [44], scientific document representations [31, 45], cascade prediction [46], and topic diffusion [47] are some of such problems that are applied on academic collaboration networks and collections of academic publications. In this section, we particularly focus on those that study the problem of trend forecasting and detection of emerging topics for academic studies.

The work in [22] challenges the problem of topic discovery and trend forecasting from texts. As in our approach, the study uses token simplification methodologies, but in a sentence level. The authors conduct association rule mining for topic discovery. Afterwards, temporal topic correlation analysis is performed and ensemble forecasting is used for topic trend prediction.

In [48], the authors focus on detecting emerging academic topics at early stage which they call *embryonic phase*. The method is based on constructing evolutionary topic co-occurence networks on a yearly basis and devising a clustering algorithm named *Advanced Clique Percolation Method (ACPM)* for detecting clusters of the topics in the evolutionary networks. After a post-processing step based on filtering among high number of clusters, the clusters of the collaborating topics that extend in increasing pace are denoted as emerging topics. The method is evaluated on a data set where the debuting topics are manually annotated.

In [49], pairwise influence between venues are studied and trending topics, which consists of topical words, are predicted for the next year's venue. Each venue is handled as a bag of words and topic embeddings are learned. Recurrent Neural Networks (RNN) over venue vectors is utilized by further considering the influence between them via the topic embeddings. The authors quantify venue to venue influence over years and detect trending topical words. Trending topics are determined by compar-

ing the word percentage increase with respect to previous years.

In [27], the authors assess paper content by extracting keywords through *deepwalk* [50] and determine popular topics in the field by examining co-occurrences of detected keywords. For keyword processing, they use the already constructed word2vec embeddings for English papers. In the study, keyword extraction is performed with a feature based approach.

The study in [28] also focuses on detecting emerging academic topics. The employed method involves constructing temporal word2vec word embeddings from a collection of academic papers belonging to a given time window and determining the increase in the ranking of keywords. The experiments are conducted on paper collections of two different venues, and the results are compared against trendy search queries and increase in citations.

In [23], the authors aim to predict research concepts that will be investigated in the next 5 years. They maintain scientific knowledge as an evolving network, called *SemNet*, where nodes represent physical concepts and edges connect pairs of topics both of which are studied in a research article. The authors employ a neural network model to rank concepts by using 17 features of the network properties. With the help of this ranking, the method suggests novel concept pairs that might be studied in the next 5 years.

The study in [24] focuses on semantic consistency of research topics while predicting the research topics. The papers are represented with one hot encoding of high frequency tokens, where each one is considered as a topic. Additionally, a unified semantic space is constructed to represent the scientific influence context of different fields. The solution employed in the study is based on the use of multiple RNNs.

The authors of the study in [25] aim to discover emerging research topics. For this purpose, a two-step approach is used such that firstly popularity scores of the topics are predicted and then emerging topics are determined among them. Each topic is represented by a set of features including term frequency (TF), inverse document frequency (IDF) and the number of unique authors participating in the topic. Neural network autoregression (NNAR) and LSTM are used as the predictive models.

In [26], scientific research topic trend prediction problem is modeled as prediction of token distributions over publication observations. Papers as well as research fields are represented with one hot encoding of tokens. Additionally, influence graph of publications is constructed by using the similarities between paper representations. The authors use Graph Convolutional Network (GCN) in addition to LSTM in their solution.

Among the related works, there are studies that use neural network based solution such as in [49], [23], [25] and [26]. However there are basic differences in problem definition and data modeling. In [49], the focus is on venue to venue influences and the use of venue and topic embeddings. The study in [23] also differs from our work in Chapter 4 since a scientific semantic network is employed and the analysis is conducted on this network. Although the problem definition given in [23] has similarity to the problem definition of our work, a wider time window of 5 years is considered in [23] . In [25], topics are represented with feature vectors where several of the features, such as Web of Science categories, are possibly extracted from external resources. In [24] and [26], the problem modeling differ from our work such that the papers and fields are represented by one hot encoding of the topics and the influence among the fields is used for predicting the topic distributions. On the other hand, in our approach we directly focus on predicting the trend of the keyword by using only the publication history of the venues.

Topic modeling and generation is similar in [18] and [25] where they start with a predefined set of topics. Then the approach in [18] uses clustering for 500 topics and the study in [25] uses statistical techniques to generate topics with foreground and background corpus. However model outputs are not compatible as the study in [18] labels only bag of word topics to increase and decrease whereas the one in [25] ranks the candidate bag of word topics. Experiments presented in [18] similarly utilize the last 5 years as an observation window. The problem definition and model outputs are similar in [19] and [26], they detect keyword set as conference topics for the next year for all conferences in the experiments. Unfortunately this modeling is not compatible with our experiments.

2.3 Document Embedding Studies

In deep learning for NLP applications, it is a common practice to utilize semantic vector space models to process tokens of a text. Deep learning models can learn representation for given tokens by starting from random state and utilizing gradient descent. Mainly word vectors are pre-trained on various general tasks with huge data sets. Tasks that depend on word embeddings can then further fine tune the pre-trained embeddings. There are several transfer learning approaches to facilitate fine tuning [51].

Word analogy task described in [52], known as *word2vec*, enables similarity and analogy calculations by vector operations. As a pre-trained embedding collection, GloVe [53] is popularly used as it aims to solve the shortcomings of locality of skip-gram based training. Main approach in GloVe is to incorporate global corpus with using global co-occurrence counts. In this way, most frequent 400,000 word tokens are trained on 42 billion token corpus.

Dynamic word2vec model is suggested in [54] to capture semantic meaning change via learning temporal word embeddings. Experiments to discover semantic trajectories are run over news dataset from NYTimes labeled with news sections. The authors consider word trends by investigating word vector norms across time.

Document vectors are also explored in our work in Chapter 4 with the main goal to query with the help of word embedding. In [30], word2vec is further extended to doc2vec in order to generate *paragraph vectors*. In [55], a similar approach to doc2vec is followed and it is aimed to simplify document vector generation by averaging word vectors. In the training phase, with the help of term frequencies, common word vector values are heavily reduced to zero to make significant words to contribute more to the resulting document vector.

Cite2vec [45] is a visualization system that learns word and referenced document embeddings through citation information via enhanced skip-gram approach. It enables users to interactively search for word and document usages to explore a research field.

Specter [31] is a deep learning transformer model specialized for scientific papers,

which uses citation information in training phase and can produce stable document embedding afterwards.

In our work in Chapter 4, we use word embedding and document embedding approaches to obtain representations of either tokens or the papers, and use them in our proposed neural architectures for trend prediction of query keyword.

CHAPTER 3

INFUENCEE ORIENTED TOPIC PREDICTION: INVESTIGATING THE EFFECT OF INFLUENCE ON THE AUTHOR

3.1 Introduction

Social network analysis is concerned with analysing the structure of the network and behaviour of individuals forming the network [5]. Although early studies in social network analysis focused on building descriptive models, with increasing amount of social network data, the research direction moved to building predictive models [6]. Such predictive models can be used in a variety of domains such as link prediction for friend recommendation [7, 8], influence detection for advertisement domain [9], and community detection for urban safety domain [10].

In this chapter [1], we work on *academic social networks* (also called *scientific collaboration networks* [11]) with a predictive point of view. In academic social networks, nodes represent entities such as authors and papers and edges represent relations such as authoring a paper and co-authorship. There are various studies in the literature on academic social networks aiming to predict collaboration patterns [12, 13, 14]. We focus on the problem of topic adoption, and propose a method to predict topic adoption of an author. More specifically, given an author and a new topic for the author, we aim to predict whether the author will publish a work on the given topic in the next time slot. Yang et al. studied this problem in [11], and propose to use two features, *topic similarity*, and *social influence*, in a regression based model. Social

[1] Murat Yukselen, Alev Mutlu, and Pinar Karagoz. 2019. "Infuencee Oriented Topic Prediction: Investigating the Effect of Influence on the Author". In Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics (WIMS2019). Association for Computing Machinery, New York, NY, USA, Article 14, 1–9. https://doi.org/10.1145/3326467.3326488 Reprinted with respect to author rights on https://authors.acm.org/author-services/author-rights

influence denotes the co-authors of the author who already adopted the given topic, and topic similarity denotes the similarity between the given topic and the topics that are already adopted by the author.

In this work, we argue that the amount of influence accumulated on the author for the given topic can affect the topic adoption. To this aim, we propose a new feature, *influencee score*, and propose an algorithm to compute this feature. The proposed algorithm is inspired from the influence detection method given in [29], in which the amount of influence going out from an author to other authors within social stream is computed. For computing influencee score, we invert the flow of the stream and find the amount of influence *towards the author* from other authors through social stream in a time line. The effect of this new feature for topic adoption prediction is analyzed within a multiple logistic regression model together with and in comparison to the features given in [11]. Additionally, we compare the performance against the baseline model of [11], as well. The experiments reveal that the proposed feature improves the prediction accuracy.

The contributions of this chapter can be summarized as follows:

- For topic adoption in scientific collaboration networks, we propose a new feature, *influencee score* (F_{InfSc}).

- For computing the *influencee score* of an author, an algorithm is proposed.

- The effect of *influencee score* for topic adoption prediction is analyzed within a multiple regression model.

The rest of the chapter is organized as follows. The background studies, which are adopted in this work, are described in Section 3.2. Section 3.3 includes the detailed description of the proposed feature and the algorithm for its computation. We present the experiments and results in Section 3.4. Finally, the overview of the work and future directions are given in Section 3.5.

3.2 Background

This study is motivated by approaches presented in [29, 11] to predict topic adoption of an author. In the subsequent sections we explain these studies in detail.

3.2.1 Efficient Influence Querying

In [29], authors propose a method to query influencers in dynamic social networks in a context-sensitive and time-aware manner. Authors assume that social stream is generated by propagating contents, represented as a bag of keywords, between users, and user a_j is influential on user a_k if there is a significant content flow from a_j to a_k. To calculate the influence score of a_j on a_k, authors propose a set of concepts as summarized below:

- Valid flow ($\mathcal{F}$): Flow of a keyword, K_i, from a_j to a_k is an ordered set of nodes, $a_j, b_1, \ldots, b_r, a_k$, such that a_j is the initiator of the keyword, there is a directed edge between b_i, b_{i+1}, for all i = 1 ... r - 1 and every node transmits the keyword to its neighbor after receiving the keyword.

- Flow duration (δt): This metric indicates time elapsed between initiation of a keyword K_i by a_j and its transmission to a_k.

- Decayed flow weight: δt indicates latency and large value of δt for K_i may indicate decay in its significance in the flow F. To incorporate this assumption into the model each flow is assigned with a weight given by $2^{-(\lambda \delta t)}$ where λ is a decay factor.

- Aggregate flow path: Aggregate flow path for keyword K_i at time t_c along a particular path, P, is the sum of weights of all valid distinct flows of the keyword along the same path. Two flows are considered distinct if the flows are initiated at different times.

- Aggregate pairwise flow: Aggregate pairwise flow between a_j and a_k for keyword K_i at time t_c is the summation of the aggregate flow paths on every path from a_j to a_k.

- Atomic influence value: Atomic influence value of a_j on a_k is the summation of all aggregate pairwise flow score of each keyword originated in a_j and transmitted to a_k.

Authors implement the above mentioned concepts on a data structure called *Flow Path Tree*, $\mathcal{T}$. In $\mathcal{T}$, root is *null* node, a path from root a leaf corresponds to a valid flow, and each node is associated with a weight corresponding to flow weight from root the that node. Algorithm 1 outlines the influencer score generation process.

Authors investigated performance of the proposed method on an academic social network. In their setting, each node corresponds to an author and edges between nodes are placed based on co-author relationship. Keyword set is constructed by extracting uni-, bi-, and tri-grams of words appearing in the title and the abstract of the papers.

3.2.2 Topic Adoption Prediction for Authors

In [11], Yang et al. investigate the topic-following behavior of researchers and propose a topic-following model to predict topic of the next publication of a researcher. Authors claim that social influence and homophily are driving factors of topic-following for a researcher. In case of scientific collaboration network, social influence corresponds to tending to adopt a topic that is most widely studied among researcher's co-authors and homophily corresponds to similarity of scientific publications. Authors also argue in their study that assigning a weight for these factors is a difficult task and build multiple regression model, with social influence score and homophily score as independent variables, to predict topic adaption. The multiple regression model is given in 3.1 where F_{SI} and F_{TS} are, respectively, special influence score and homophily score of an author.

$$logit[\pi(x)] = \alpha + \beta_1 F_{SI} + \beta_2 F_{TS} \tag{3.1}$$

Social influence is calculated according to 3.2 where $F_{SI}(u, s, t)$ indicates the probability of researcher u will follow topic s in year t. In 3.2, $N'(u)$ indicates co-authors of u who published on topic s before u, $w(e_{u,v})$ is the weight of the edge between u

Algorithm 1 Original UpdateFlowPaths algorithm

function UPDATEFLOWPATHS($a_i, G(t), \mathcal{S}, \mathcal{T}$) ▷ a_i: Originating Node, $G(t)$: Network, $\mathcal{S}$: Social Stream, $\mathcal{T}$: Flow-Path Tree
  Receive the next message containing keyword K in social stream $\mathcal{S}$ originating at node a_i
  Create singleton node a_i in tree $\mathcal{T}$ as child root of node if it does not already exists
  $C \leftarrow a_i$ ▷ Set of candidate paths for expansion
  Update weight of singleton path containing only node a_i in tree $\mathcal{T}$ by 1
  while C is not empty **do**
    Delete the first path P from C and denote the first node of P by a_j
    for each $a_k \notin P$ in $V(t)$ with an incoming edge to a_j **do**
      if prefix of path $a_k \bigoplus P$ exists in $\mathcal{T}$ and a_k has propagated keyword K prior to a_j **then**
        if the complete path $a_k \bigoplus P$ exists in $\mathcal{T}$ **then**
          Increment weight of last node of path $a_k \bigoplus P$ by 1 in $\mathcal{T}$
        else
          Create last node of P as child for prefix of path $a_k \bigoplus P$ in $\mathcal{T}$ with weight as 1
        end if
        Add $a_k \bigoplus P$ to C
      end if
    end for
  end while
end function

and v and $f(v, s, t-1)$ indicates the influence from u's neighbor v in $t-1$.

$$F_{SI}(u, s, t) = \sum_{v \in N'(u)} \frac{w(e_{u,v})}{\sum_{v \in N'(u)} w(e_{u,v})} \times f(v, s, t-1) \tag{3.2}$$

Homophily score of an author, u, is calculated according (3.3). Topic similarity of u's and his/her co-authors' publications, where u' is aggregated paper counts of u and $U^s_{<t}$ is the aggregated paper counts up to time t. Topic similarity of two authors, u and v, is calculated based on cosine similarity given in (3.4) where u and v are vectors and v_i indicates number of publications by v in the i^{th} topic.

$$F_{TS}(u, s, t) = sim(u', U^s_{<t}) \tag{3.3}$$

$$sim(u, v) = cosine(u, v) = \frac{u \cdot v}{\| u \| \| v \|} \tag{3.4}$$

To evaluate their model, authors consider publications in three-years intervals. To predict topic of a paper published in year t, papers published in years $[t-3, t-1]$ are considered for feature calculations and observation is made in year $[t, t+2]$ interval.

3.3 Proposed Method: Incorporating Influence Factor in Topic Prediction

In this study we propose a new feature, influence score, and present a model that incorporates it together with social influence, and topic similarity. Effect of social influence and topic similarity in topic adoption are discussed in [11] and are implemented in a similar fashion in this study.

Incorporating influencee score in topic adoption is motivated by [29]. However, ideas presented in [29] can not be applied directly as [29] provides mechanisms to capture influencer score, i.e. $I(u, *, keywords(s), t)$, while we are interested in influencee score which is formulated in Equation (3.5). In this equation, u represents an author, s represents a topic and t represent time. Function $I(\cdot)$ calculates the accumulated influence on user u for topic s, represented by a set of keywords *keywords(s)*, at time t.

$$F_{InfluenceeScore}(u, s, t) = I(*, u, keywords(s), t) \tag{3.5}$$

Although the data structure to represent scientific collaboration network presented in [29], namely *Flow Path Tree* - $\mathcal{T}$, forms basis for influencee score calculation, it can not be directly adopted in this study. In $\mathcal{T}$, influencers appear under nodes representing keywords and ranked influencers appear 2 levels down from the virtual root node and for efficiency issues nodes at deeper levels are pruned. In this study, we are interested in influencee score and nodes representing influencees appear at deeper levels of the tree, ideally at leaves. Hence, such a pruning mechanism avoids representation of influencees. In order to overcome this limitation, the flow path tree is held in reverse order and the modified algorithm consumes social stream backwards. Figure 3.1 illustrates the modified data structure where directed arrows between authors (w_y, w_u) and (w_v, w_u) denote influence propagation aligned with time as it becomes more recent near to influencee node w_u under keyword node k_1. Author nodes have influence weights and timestamp for their last update. In that sense author w_u will not have any weight since we are interested in how much influence it receives.

When an influence emerges, influencer node receives weight for the timestamp. The update incorporates decay in a way that when a timestamp changes, here we are receiving older messages and timestamp decreases, weight is decayed according to time delta and new weight is added. Delta of the timestamps between nodes is shown as distance between author nodes in Figure 3.1. At step 1 for $Stream_{t-1}$, author v and y publish a paper at $t-1$ and because of their author graph G_{t-1} it is a valid influence. At step 2 for $Stream_{t-2}$, it is understood that author y has influenced before, so node w_y's weight is decayed and it is displaced to $t-2$.

In Algorithm 2, τ tree is initialized layer by layer as virtual root node, keyword nodes, $author_u$ with zero weight, $author_u$'s friends $author_v$ with zero weight. Reversed stream of papers are then used to construct τ that holds information specifically to answer how much influence $author_u$ received. Updating the flow paths is described in algorithm 3.

Influence is calculated as an accumulation of all paths from keyword nodes to leaves where path score is summed by decayed weight with respect to author node times-

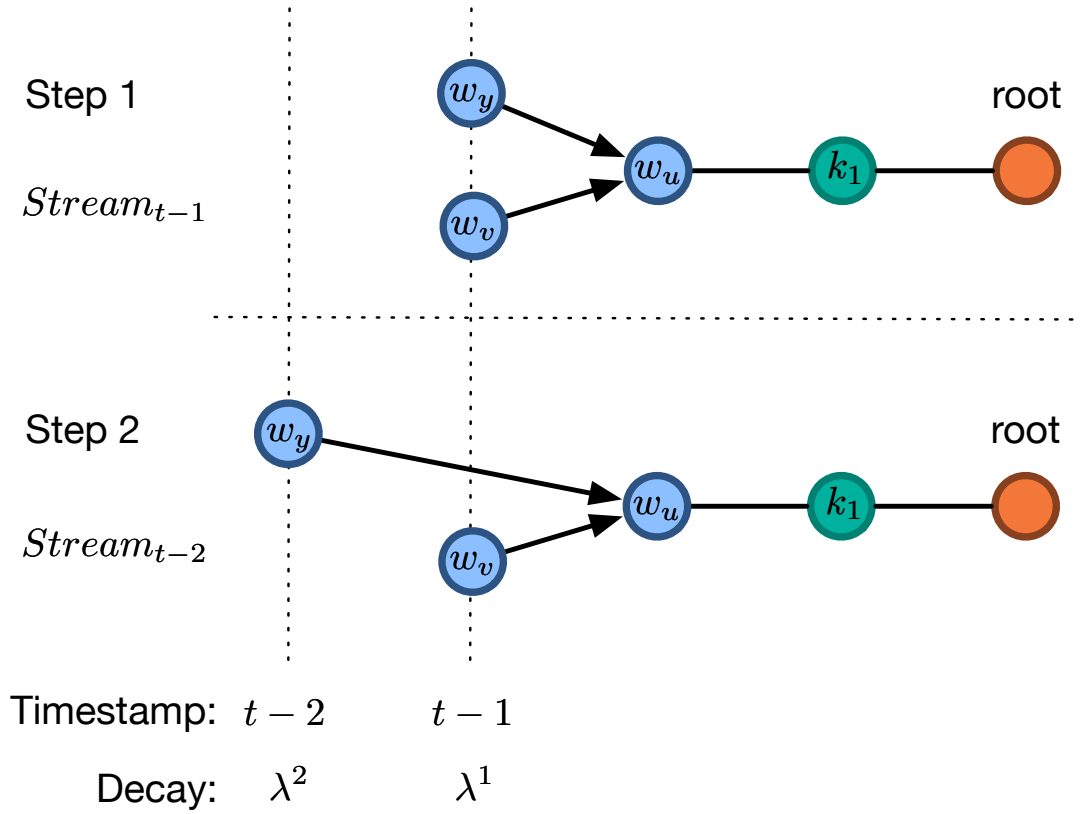


Figure 3.1: Example flowpath tree

tamp. The effect of decay is similar to Katz measure [56]. This aggregation process is given in Algorithm 4.

3.4 Experiments

3.4.1 Data Sets and Experimental Setting

To evaluate performance of the proposed topic adoption model a set of experiments is conducted on data retrieved from Arnet Miner [57] and Microsoft Academic Graph (MAG) [58].

MAG database is used retrieve papers to build a SVM to predict topic of a paper. To this aim, we retrieved top 10 papers belonging to topics of algorithm and complexity, classification, information retrieval, privacy and security, and text and web mining. We retrieved uni-, bi-, and trigrams from titles and abstracts of the papers and used

Algorithm 2 Influencee score f_{InfSc}

function FEATUREINFSC(u, s, t) ▷ u: author, s: topic, t: time
  $S_{reversed} \leftarrow$ filter the social stream S up to time t with keywords for s
  $\tau \leftarrow$ a flow-path-tree for author u with all u's neighbors are expanded with weight 0
  $score \leftarrow 0$
  for all p in $S_{reversed}$ **do** ▷ p: paper as message
    for all k in $p.filteredKeywords$ **do**
      for all v in $p.authors$ **do**
        updateFlowPathTree(τ, k, v, t_p) ▷ t_p: time of p
      end for
    end for
  end for
  return $aggregatedScore(\tau, t)$ ▷ $Influence(*, author_u, *, t)$ as in [29]
end function

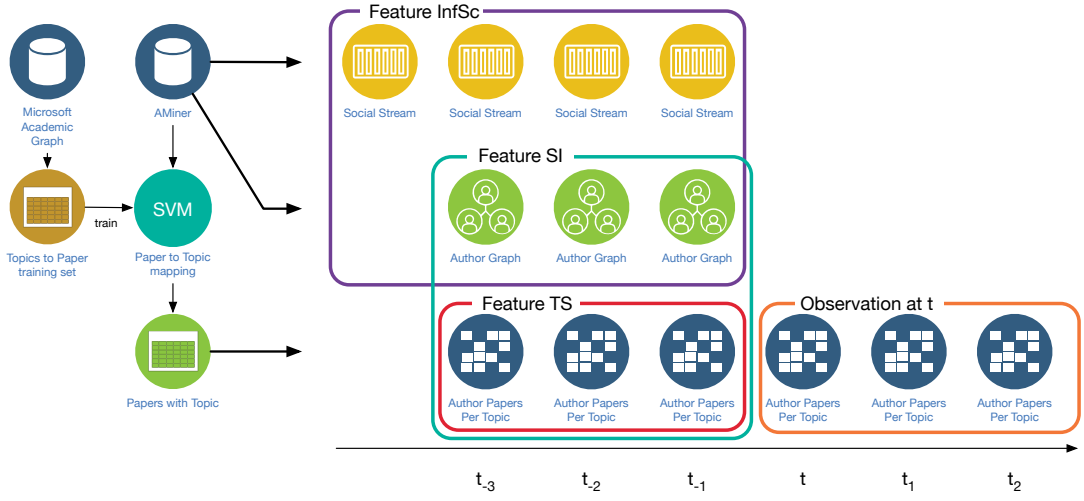


Figure 3.2: Data flow of the features

these keywords to train SVM.

From Arnet Miner database we retrieved papers published between 2001 and 2011 and assigned their topic using our SVM model. In Table 3.1 we list the number of papers retrieved from Arnet Miner [57] and give yearly tagged papers in their respective columns.

Algorithm 3 UpdateFlowPaths function

1: **function** UPDATEFLOWPATHTREE(τ, k, v, t)
2: Starting from keyword node k, traverse all open
3: **for all** author $node_w$ as direct friend of influencee **do**
4: **if** $node_w$ is v **then**
5: Increment weight incorporating decaying and return
6: **end if**
7: **for all** child $node_x$ of $node_w$ **do**
8: $dfsPathUpdate(node_x, v, t)$
9: **end for**
10: **end for**
11: **end function**
12: **function** DFSPATHUPDATE($node_x, v, t$)
13: **if** $t > t_{node_x}$ **then return**
14: **else if** $node_x$ is v and t is t_{node_x} **then**
15: increment weight incorporating decay and return
16: **else if** v is friend of $node_x$ **then**
17: append $node_v$ to $node_x$ with decayed weight and return
18: **end if**
19: **for all** already open $node_y$ of $node_x$ **do** ▷ these friends already conveyed information to influencee
20: recurse with $dfsPathUpdate(node_y, v, t)$
21: **end for**
22: **end function**

Algorithm 4 Aggregate Influencee score from τ

function AGGREGATEDSCORE(τ, t) ▷ τ: flow-path tree, t: time
  $score \leftarrow 0$
  for all $node_k$ in τ **do**
    for all path p starting from $node_k$ **do**
      for all $node_a$ in p **do**
        add $weight_{node_a} * decay(t, t_{node_a})$ to $score$
      end for
    end for
  end for
  return $score$
end function
function DECAY(t_1, t_2) **return** $2^{(-decayFactor*(t_1-t_2)}$
end function

Although papers have topics, in our model network is based on authors and keywords. For this purpose, for each topic we build a keyword dictionary that contains top 30 terms of the topic. To query the stream for a specific topic we use the topic's keyword dictionary. The statistics of the filtered stream is given in Table 3.2.

Figure 3.2 illustrates our experimental setting. The leftmost part visualizes the data preparation step, data gathering and SVM training. The right part of the Figure 3.2 visualizes model training and prediction steps. For an author, say u, that has not published any paper on topic s, his/her stream for the past three years is queried with $keyword(s)$ which is made up of top 30 keywords on the topic. Influence Score feature needs yearly social streams and author graphs in computation. Similarly, Feature Social Influence makes use of Author Graphs and yearly Author Paper Counts per Topic. Feature Topic Similarity is simply calculated based on the cosine similarity of user u with respect to all users on topic s up to time t. Observation result is a simple lookup in 3 consecutive years whether u has a publication on that topic or not.

In order to analyze the effect of the features, various models are trained with Multiple Logistic Regression having different set of features. Each model name is set such that it expresses the set of features used. In that respect, *mlr* as a short name was used

Table 3.1: Yearly paper counts per topic

Year	**Total**	**Alg. C.**	**Class.**	**IR**	**P.& Sec.**	**W.& T. M.**
2001	56783	19464	1678	3990	2448	5833
2002	62029	20262	2096	4458	2903	6358
2003	54874	18184	1900	3913	2528	5672
2004	61169	19635	2315	4402	3279	6034
2005	98203	31166	5036	6575	5591	8539
2006	121189	41547	6246	7991	5896	9894
2007	116194	36312	6496	7913	5778	10448
2008	128681	38931	7878	8754	6029	11124
2009	169575	50779	11545	10381	6967	13764
2010	139357	42951	8446	9060	5653	12555
2011	131262	41350	7553	8556	5078	12355

Table 3.2: Social Stream Paper Counts per Year for Influencee Score Calculation

Year	**count**
2001	35447
2002	48473
2003	46550
2004	52027
2005	87755
2006	108749
2007	106364
2008	121537
2009	162249
2010	133040
2011	124158

in [11] as a model with two features: F_{SI} and F_{TS}. This work focused on adding a third feature as $F_{InfluencesScore}$ (F_{InfSc}). The experiments are conducted with a balanced number of samples for training and test, as given in Table 3.3. The numbers

Table 3.3: Training and test sample counts per topic

	Alg. Complex.	**Class.**	**Inf. Retrvl.**	**Prv. & Sec.**	**Web & Text Mining**
train	4043	4090	4115	4174	4153
test	1991	2067	2031	2034	2077

of positive and negative samples are balanced as well.

3.4.2 Experimental Results

For the selected topics, the parameters of each model are given in Tables 3.4, 3.5, 3.6, 3.7, and 3.8 respectively.

Table 3.4: Model parameters of Alg. and Complex. Topic

Model	**Feature**	**Par.**	**Value**	**Std.Err.**	**Wald**	**sig.**
$\boldsymbol{F_{CE}}$	**intercept**	α	0.0427	0.0042	10.0701	0
	F_{CE}	β_1	0.7610	0.0014	532.3658	0
$\boldsymbol{F_{SI} + F_{TS}}$	**intercept**	α	0.0232	0.0040	5.6945	6.2524e-09
	F_{SI}	β_1	-0.0098	0.0078	-1.2601	0.1038
	F_{TS}	β_2	3.7651	0.0074	506.5743	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	**intercept**	α	-1.3639	0.0111	-121.9350	0
	F_{SI}	β_1	-0.0170	0.0242	-0.7022	0.2412
	F_{TS}	β_2	3.8134	0.0240	158.7485	0
	F_{InfSc}	β_3	0.0853	0.0109	7.8076	3.6637e-15
$\boldsymbol{F_{TS}}$	**intercept**	α	0.0158	0.0029	5.3932	3.4766e-08
	F_{TS}	β_1	3.8065	0.0055	688.4928	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	**intercept**	α	-0.0411	0.0110	-3.7387	9.3618e-05
	F_{SI}	β_1	0.0127	0.0308	0.4128	0.3398
	F_{InfSc}	β_2	0.1062	0.0145	7.3054	1.6220e-13
$\boldsymbol{F_{TS} + F_{InfSc}}$	**intercept**	α	-1.3688	0.0081	-167.9981	0
	F_{TS}	β_1	3.8420	0.0179	214.6297	0
	F_{InfSc}	β_2	0.0440	0.0069	6.3233	1.3570e-10

Table 3.5: Model parameters of Classification Topic

Model	Feature	Par.	Value	Std.Err.	Wald	sig.
$\boldsymbol{F_{CE}}$	**intercept**	α	-2.0091	0.0029	-685.3135	0
	F_{CE}	β_1	1.0696	0.0022	465.7863	0
$\boldsymbol{F_{SI} + F_{TS}}$	**intercept**	α	-2.5683	0.0046	-553.9986	0
	F_{SI}	β_1	0.0879	0.0184	4.7673	9.3854e-07
	F_{TS}	β_2	3.6437	0.0125	291.2114	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	**intercept**	α	-1.3941	0.0157	-88.3621	0
	F_{SI}	β_1	0.0737	0.0669	1.1013	0.1353
	F_{TS}	β_2	3.1679	0.0394	80.3719	0
	F_{InfSc}	β_3	0.0002	7.7684e-05	2.6408	0.0041
$\boldsymbol{F_{TS}}$	**intercept**	α	-2.5617	0.0034	-744.2074	0
	F_{TS}	β_1	3.6426	0.0093	390.2773	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	**intercept**	α	-0.0097	0.0100	-0.9700	0.1660
	F_{SI}	β_1	-0.0777	0.0676	-1.1488	0.1253
	F_{InfSc}	β_2	0.0012	8.0652e-05	16.0311	0
$\boldsymbol{F_{TS} + F_{InfSc}}$	**intercept**	α	-1.5236	0.0116	-130.9340	0
	F_{TS}	β_1	3.3502	0.0291	114.9896	0
	F_{InfSc}	β_2	0.0004	5.5377e-05	7.3600	1.0191e-13

Table 3.6: Model parameters of Information Retrieval Topic

Model	Feature	Par.	Value	Std.Err.	Wald	sig.
$\boldsymbol{F_{CE}}$	**intercept**	α	-1.4699	0.0038	-377.0467	0
	F_{CE}	β_1	0.9033	0.0021	419.0485	0
$\boldsymbol{F_{SI} + F_{TS}}$	**intercept**	α	-1.5351	0.0064	-237.1857	0
	F_{SI}	β_1	-0.1394	0.0226	-6.1706	3.4505e-10
	F_{TS}	β_2	3.0390	0.0177	170.9132	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	**intercept**	α	-1.2779	0.0173	-73.5049	0
	F_{SI}	β_1	-0.1334	0.0647	-2.0618	0.0196
	F_{TS}	β_2	2.9658	0.0471	62.8816	0
	F_{InfSc}	β_3	0.3001	0.0131	22.8290	0
$\boldsymbol{F_{TS}}$	**intercept**	α	-1.5915	0.0047	-331.7382	0
	F_{TS}	β_1	3.1183	0.0132	234.7923	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	**intercept**	α	-0.0509	0.0104	-4.8885	5.2544e-07
	F_{SI}	β_1	-0.1832	0.0652	-2.8073	0.0025
	F_{InfSc}	β_2	0.3852	0.0136	28.1625	0
$\boldsymbol{F_{TS} + F_{InfSc}}$	**intercept**	α	-1.3271	0.0127	-103.7753	0
	F_{TS}	β_1	2.9698	0.0346	85.7041	0
	F_{InfSc}	β_2	0.3566	0.0083	42.8263	0

Table 3.7: Model parameters of Privacy and Security Topic

Model	Feature	Par.	Value	Std.Err.	Wald	sig.
$\boldsymbol{F_{CE}}$	**intercept**	α	-1.9830	0.0032	-610.0604	0
	F_{CE}	β_1	1.0433	0.0019	526.6821	0
$\boldsymbol{F_{SI} + F_{TS}}$	**intercept**	α	-2.5836	0.0053	-486.2342	0
	F_{SI}	β_1	-0.0357	0.0168	-2.1268	0.0167
	F_{TS}	β_2	4.5032	0.0153	294.2476	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	**intercept**	α	-1.7229	0.0164	-104.9034	0
	F_{SI}	β_1	0.1734	0.0540	3.2111	0.0006
	F_{TS}	β_2	4.0029	0.0436	91.7938	0
	F_{InfSc}	β_3	1.7738e-05	1.2257e-06	14.4709	0
$\boldsymbol{F_{TS}}$	**intercept**	α	-2.5612	0.0039	-648.7085	0
	F_{TS}	β_1	4.4228	0.0114	385.7505	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	**intercept**	α	4.3715e-10	0.0106	4.1060e-08	0.4999
	F_{SI}	β_1	1.0555e-09	0.0591	1.7842e-08	0.4999
	F_{InfSc}	β_2	2.3117e-05	1.3952e-06	16.5688	0
$\boldsymbol{F_{TS} + F_{InfSc}}$	**intercept**	α	-1.3728	0.0123	-111.5988	0
	F_{TS}	β_1	3.1942	0.0328	97.1137	0
	F_{InfSc}	β_2	1.3715e-05	7.5793e-07	18.0959	0

Table 3.8: Model parameters of Text and Web Mining Topic

Model	Feature	Par.	Value	Std.Err.	Wald	sig.
$\boldsymbol{F_{CE}}$	**intercept**	α	-1.3166	0.0040	-324.8517	0
	F_{CE}	β_1	0.8476	0.0021	390.1226	0
$\boldsymbol{F_{SI} + F_{TS}}$	**intercept**	α	-1.4135	0.0063	-221.7997	0
	F_{SI}	β_1	-0.1682	0.0226	-7.4396	5.2069e-14
	F_{TS}	β_2	3.0144	0.0171	175.7761	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	**intercept**	α	-1.2157	0.0160	-75.7983	0
	F_{SI}	β_1	-0.3031	0.0597	-5.0707	2.0679e-07
	F_{TS}	β_2	2.9363	0.0427	68.6542	0
	F_{InfSc}	β_3	4.5301e-05	8.2357e-07	55.0053	0
$\boldsymbol{F_{TS}}$	**intercept**	α	-1.4237	0.0047	-300.9280	0
	F_{TS}	β_1	3.0240	0.0128	235.7353	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	**intercept**	α	-9.0498e-09	0.0098	-9.1708e-07	0.4999
	F_{SI}	β_1	-2.9751e-09	0.0599	-4.9587e-08	0.4999
	F_{InfSc}	β_2	5.7770e-05	8.4983e-07	67.9780	0
$\boldsymbol{F_{TS} + F_{InfSc}}$	**intercept**	α	-1.2412	0.0120	-102.8170	0
	F_{TS}	β_1	2.9433	0.0324	90.5676	0
	F_{InfSc}	β_2	4.2191e-05	6.7121e-07	62.8586	0

In the model parameter tables, Wald and significance values help to interpret the feature contributions such that positive Wald value is considered worthy feature for the model at hand. Similarly parameters with significance value smaller than 0.05 are also useful. In light of this information, model parameters can be interpreted as follows: F_{TS} is the most effective feature and F_{InfSc} follows in second place.

Accuracy of the models per topic are given in Tables 3.9, 3.10, 3.11, 3.12, and 3.13 respectively. In terms of F_β scores, proposed "$F_{SI} + F_{TS} + F_{InfSc}$" model and the feature F_{InfSc} performed better in 4 out of 5 topics selected. $\beta = 1.1$ is used similar to [11] favoring recall performance. Although Receiving Operational Characteristic (ROC) [59] value is the highest in Algorithm and Complexity topic (Figure 3.3(a)), F_β score evaluates our model as subpar.

Table 3.9: Model performance of Algorithms and Complex.

model	**recall**	F_β	**acc.**	**prec.**	**spec.**
$\boldsymbol{F_{CE}}$	1	0.8845	0.7761	0.7761	0
$\boldsymbol{F_{SI} + F_{TS}}$	1	0.8943	0.7929	0.7929	0
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	0.7136	0.7372	0.7368	0.7680	0.7071
$\boldsymbol{F_{TS}}$	1	**0.8949**	0.7940	0.7940	0
$\boldsymbol{F_{SI} + F_{InfSc}}$	0.2235	0.3091	0.5299	0.5759	0.5188
$\boldsymbol{F_{TS} + F_{InfSc}}$	0.6981	0.7273	0.7297	0.7662	0.6962

Table 3.10: Model performance of Classification

model	**recall**	F_β	**acc.**	**prec.**	**spec.**
$\boldsymbol{F_{CE}}$	0.4805	0.5042	0.7177	0.5362	0.7850
$\boldsymbol{F_{SI} + F_{TS}}$	0.3019	0.4212	0.7629	0.8071	0.7571
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	0.7265	**0.6695**	0.6279	0.6114	0.6529
$\boldsymbol{F_{TS}}$	0.3052	0.4247	0.7653	0.8074	0.7598
$\boldsymbol{F_{SI} + F_{InfSc}}$	0.4475	0.5084	0.5755	0.6087	0.5558
$\boldsymbol{F_{TS} + F_{InfSc}}$	0.7111	0.6631	0.6302	0.6131	0.6541

ROC curves per topic are shown in Figure 3.3. As seen in the results *Coauthor effect* feature performs better for Privacy & Security topic, Text & Web Mining and Infor-

Table 3.11: Model performance of Information Retrieval

model	recall	F_β	acc.	prec.	spec.
$\boldsymbol{F_{CE}}$	0.6845	0.6442	0.6850	0.6015	0.7579
$\boldsymbol{F_{SI} + F_{TS}}$	0.6124	0.5724	0.5953	0.5305	0.6608
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	0.7898	**0.6902**	0.6120	0.5988	0.6416
$\boldsymbol{F_{TS}}$	0.6158	0.5709	0.5929	0.5245	0.6633
$\boldsymbol{F_{SI} + F_{InfSc}}$	0.3813	0.4742	0.5992	0.6724	0.5703
$\boldsymbol{F_{TS} + F_{InfSc}}$	0.7962	**0.6927**	0.6154	0.5986	0.6534

Table 3.12: Model performance of Privacy and Security

model	recall	F_β	acc.	prec.	spec.
$\boldsymbol{F_{CE}}$	0.6180	0.5751	0.7263	0.5304	0.8287
$\boldsymbol{F_{SI} + F_{TS}}$	0.4116	0.5017	0.7581	0.6827	0.7754
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	0.8193	**0.7007**	0.6219	0.5963	0.6834
$\boldsymbol{F_{TS}}$	0.3983	0.4921	0.7570	0.6882	0.7721
$\boldsymbol{F_{SI} + F_{InfSc}}$	1	0.6859	0.4970	0.4970	0
$\boldsymbol{F_{TS} + F_{InfSc}}$	0.8030	0.6967	0.6200	0.6006	0.6642

Table 3.13: Model performance of Text and Web Mining

model	recall	F_β	acc.	prec.	spec.
$\boldsymbol{F_{CE}}$	0.7433	0.6628	0.6645	0.5861	0.7581
$\boldsymbol{F_{SI} + F_{TS}}$	0.6644	0.6057	0.6014	0.5473	0.6666
$\boldsymbol{F_{SI} + F_{TS} + F_{InfSc}}$	0.8259	**0.7001**	0.6186	0.5912	0.6875
$\boldsymbol{F_{TS}}$	0.6725	0.6105	0.6037	0.5493	0.6708
$\boldsymbol{F_{SI} + F_{InfSc}}$	0.8953	0.6940	0.5793	0.5456	0.7249
$\boldsymbol{F_{TS} + F_{InfSc}}$	0.8395	**0.7095**	0.6245	0.5975	0.6961

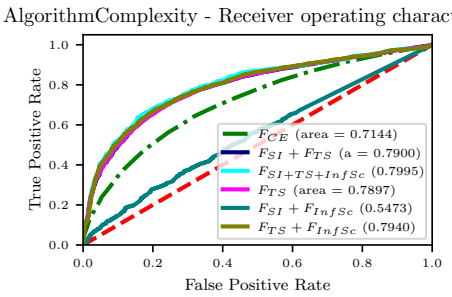
mation Retrieval topics. It is simply calculated as logarithm of one plus the number of already active friends on the topic. Although this feature is performing very good with respect to its simplicity, it is not taking into account the time aspect of author interests and does not incorporate any decay and further links in the social network. Proposed model including the feature *Influencee Score* (F_{InfSc}) performs better in all ROC curves.

3.5 Discussion

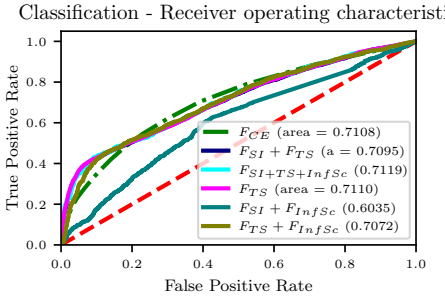
In this chapter, we investigate the effect of accumulated influence for topic adoption prediction in a scientific collaboration network. We use the term *influencee score* to denote the influence accumulated on an author for a given topic. We hypothesize that the influence accumulated on an influencee for a topic may be effective on topic adoption. The influencee score calculation method is inspired from influencer detection work in [29]. We modify the influencer score calculation such that the social stream is played backwards in order to accumulate the influence coming from various sources. We incorporate the influencee score as a feature within the framework presented in [11].

The experiments conducted on Arnet Miner data set show that the proposed feature, F_{InfSc}, improves the prediction performance, especially recall value, when used together with the features presented in [29]. The prediction performance is better when F_{InfSc} is used together with F_{TS} only. Another interesting observation is that, on the contrary to results reported in [11], F_{CE} provides a good performance for most of the topics especially in terms of specificity.

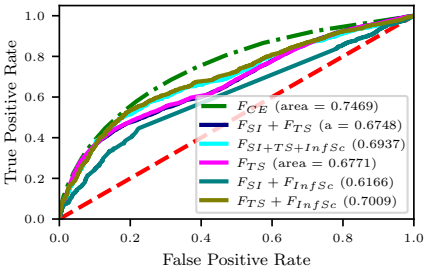
As a future work, further feature combinations can be explored to improve the prediction performance. Another interesting enhancement direction is extending the scientific collaboration network with additional elements such as publication venues.



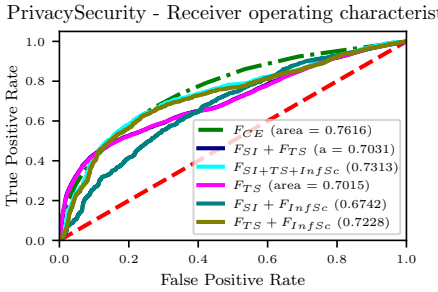
(a) Algorithm Complexity



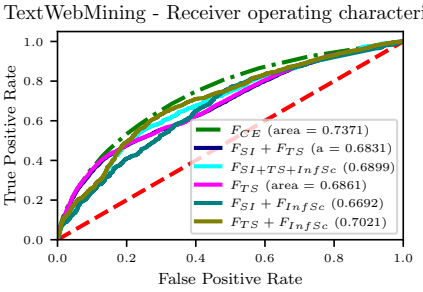
(b) Classification



(c) Information Retrieval



(d) Privacy and Security



(e) Text and Web Mining

Figure 3.3: Receiving Operational Characteristic for all topics

CHAPTER 4

PREDICTING THE TRENDING RESEARCH TOPICS BY DEEP NEURAL NETWORK BASED CONTENT ANALYSIS

4.1 Introduction

Predicting trends has always been an attractive ability in different domains. From fashion to social media, analyzing the current condition and predicting the future behavior brings advantages such as being the first or early adopter of the trend. Furthermore, trend prediction can help policymakers to implement necessary actions. Numerous research to predict trends has been conducted on several domains such as financial markets [1, 2], public health issues [3], and environmental issues [4].

In academic research, data analytics and prediction techniques have been used in a variety of different problems such as publication prediction, collaboration analysis and academic team formation [15, 16, 17]. As another important problem, predicting the trend of the topics has several benefits for academic research community. Such insights may help funding agencies to optimize their policies [18] and guide technology companies to shape their policies [19]. In addition, knowing future research trends may help new researchers to plan their studies [20].

With the recent advancements in machine learning, deep learning based solutions have been devised for prediction problems in variety of domains [21], also successful results for various text mining and information extraction problems using recent deep neural models have been published. Academic research topic trend prediction problem has been studied earlier, but mostly through more conventional data mining and machine learning approaches [22, 19]. There are recent efforts focusing on research topic prediction [23, 24, 25] and topic trend prediction [26], but they rely on

hand-crafted features and additional information such as a semantic network, citation information or influence among research fields and venues. There are also related studies in the literature to detect hot topics [27] rather than trend prediction.

In this chapter[1], we aim to explore whether deep neural architectures can be more effective in coping with topic trend prediction problem without using hand-crafted features and external information. To this aim, we propose a set of novel neural models that use only the paper collections for processing. In the first three architectures we focus on employing word representations within different neural architectures. In the next three proposed models, we explore the use of different paper representations within the proposed neural architecture.

We formulate the challenged problem, trend detection of academic research topics, as predicting the trend of the keywords that describe topics. More specifically, we assume that a research topic can be described as a set of keywords, and we aim to predict the trend of a keyword. This assumption has been also used in related studies such as [27], [28], and also topic modeling studies in general. We model the trend prediction problem as a *supervised learning* problem with three labels such that given a keyword, we aim to predict as to whether its use will *increase*, *decrease* or will stay *steady*.

One can consider various ways to set the labels for increasing, decreasing, and steady use of keywords. We define this behavior of trend in terms of frequency distributions. For a given time window, the label is determined based on the past observation of the frequency distribution of the keyword. More specifically, we can informally define the keyword trend prediction problem as follows: Given a sequence of published papers in temporal order for a venue and a query keyword, the aim is to predict the trend label for the query keyword for the future time window.

The proposed neural architectures base on generating summary representations of the observed publications (in the observation window) in order to generate trend prediction of the query keyword (for the prediction window). Therefore input to the models is paper collections and a query keyword, and output is a trend label prediction. Since

[1] Murat Yukselen, Alev Mutlu and Pinar Karagoz, "Predicting the Trending Research Topics by Deep Neural Network Based Content Analysis," in IEEE Access, vol. 10, pp. 90887-90902, 2022, doi: 10.1109/ACCESS.2022.3202654. Reprinted with respect to author rights of IEEE Access.

the textual data is represented as a sequence of tokens, Long-Short Term Memory (LSTM) neural model is used a core component of the architectures. However it is combined with other modules (including other LSTM modules) in a novel setting for the focused prediction problem.

As stated earlier, we propose two groups of architectures on the basis of using word embedding or paper embedding in the processing. Within the first group, we propose three architectures, exploring the use of only year based summary (in Model 1), the use of both year based and observation window summary (in Model 2), and the use of both summaries where year based summary is constructed by a convolution layer (Model 3). In the second group of neural architectures, we use year based and observation based summary representations, bu this time, explore the use of different paper embedding modules, LSTM based paper embedder module (Model 4), doc2vec [30] (Model 5) and Specter [31] (Model 6).

We analyze the performance of the proposed methods on a collection of academic papers from several well-known conferences along a timeline of 13 years. The analysis is conducted per venue for a collection of test query keywords. The selected conferences have overlapping focus, but also each has its own theme and academic community. Therefore, by venue based analysis we aim to predict the trend within each theme and community. Additionally, we conduct trend prediction analysis by combining the paper collection of all the venues. This analysis provides insight about the trend in a broader research field. By this way, we additionally explore the prediction performance under a higher volume of publications and more evidence for the trend.

The contribution of this chapter can be listed as follows:

- The research trend prediction analysis is formally defined through keyword trend prediction over a history of published papers.
- A family of neural models is proposed for the defined problem. Each of the proposed neural models explore alternative ways to generate and use representations of papers and paper collections with respect to the query keyword.
- The prediction performance of the models are analyzed in comparison to regres-

sion and SVM based baseline techniques through a rich collection of academic papers from 10 venues. The analysis also includes experiments on combined paper collection of all venues to observe the trend prediction for a broader research area and under a higher volume of paper collection.

- A qualitative analysis is given to present the trend predictions for query keywords.

The chapter is organized as follows. In Section 4.2, preliminary information for the proposed method is given. In Section 4.3, the problem definition and the proposed methods are described. In Section 4.4, experiment setting and comparative prediction performance analysis of the methods are presented. The chapter is concluded in Section 4.5 with an overview of the work and the results, as well as the future work.

4.2 Preliminaries and Problem Definition

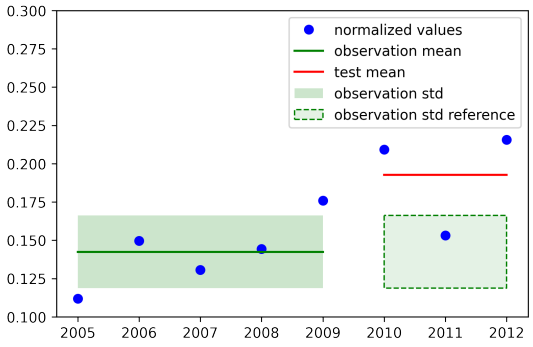
In this section, we define the basic concepts related to the problem and then give the problem definition.

Definition 4.2.1 (Query Keyword) Query keyword q *is a term or a token that has significance for a research topic.*

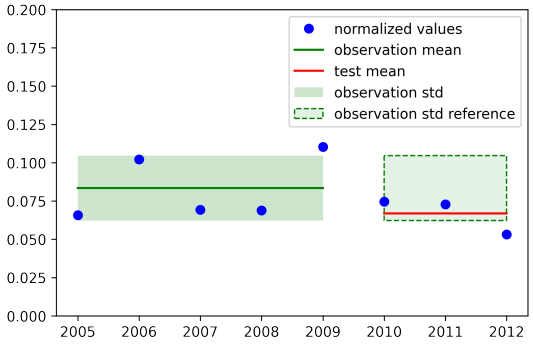
Definition 4.2.2 (Popularity) Popularity *of a query keyword* q *for a given period is the normalized frequency in a given time period. The normalization is achieved through the cardinality of the set of papers* P *within the time period, denoted as* $popularity(q, P)$.

In our work, we consider the time period as one year. Hence we calculate the normalized frequency of a query keyword per year.

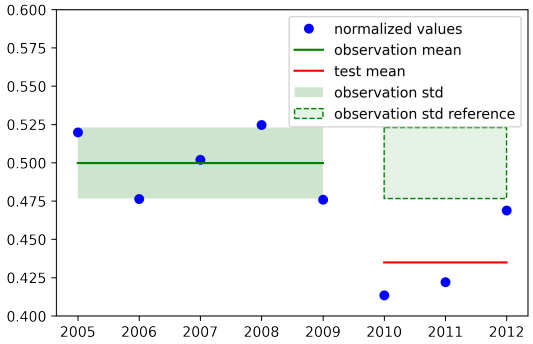
Definition 4.2.3 (Observation window) Observation Window *is a time range consisting of certain number of consecutive time periods. The paper collection for an observation window of length* l *is a list of paper collections* $< P_1, ..P_l >$*, where each* P_i *denotes a paper collection for the time period* i.



(a) Increase



(b) Steady



(c) Decrease

Figure 4.1: Illustration of the Keyword Trend Labels. First 5 years is Observation window and last 3 years is Prediction window.

In our work, the observation window length is set as 5, on the basis of validation analysis, and hence our observation window is 5 consecutive years.

Definition 4.2.4 (Prediction Window) Prediction window *is a time range consisting*

of a certain number of consecutive time periods that follow the observation window. The paper collection for a prediction window of length k is a list of paper collections $< P'_1, ..P'_k >$, where each P'_j denotes a paper collection for the time period j.

In our work, the prediction window length is set as 3 on the basis of validation analysis. This expresses that the trend label prediction is made for the subsequent 3 years.

Now that we have defined the observation and the prediction windows, trend prediction is made simply by identifying whether the popularity distribution of a given query keyword for the prediction period will be either inside, below, or above the distribution for the observation period.

Definition 4.2.5 (Observation Period Distribution) Observation period distribution *denotes the population value range $avg_{obs} \pm std_{obs}$. We define avg_{obs} and std_{obs} as follows. Given an observation window of length l and corresponding paper collection $< P_1, ..P_l >$, and a query keyword q, for each P_i, we have a popularity value $popularity(q, P_i)$. We calculate the average and standard deviation of the popularity values for the observation period, denoted as avg_{obs} and std_{obs}, respectively.*

Definition 4.2.6 (Query Keyword Labeling) Query keyword labeling *consists of choosing one of the labels* Increase*,* Steady *and* Decrease*. Given the prediction period with length k and corresponding paper collection $< P_1, ..P_k >$, avg_{pred} denotes the average popularity of the query keyword q for the prediction period. The label* Increase *denotes that $avg_{pred} > (avg_{obs} + std_{obs})$. Similarly,* Decrease *denotes that $avg_{pred} < (avg_{obs} - std_{obs})$, and* Steady *expresses that avg_{pred} lies within observation period distribution.*

Figure 4.1 illustrates the trend labels on a sample case. In Figure 4.1a, the trend is labeled as *Increase* since avg_{pred} lies above the observation distribution. Similarly, Figure 4.1b illustrates the *Steady* label and Figure 4.1c illustrates the *Decrease* label. Labels *Decrease*, *Steady* and *Increase* are mapped to -1, 0, 1, respectively for the regression model . Classification models assign the values $0, 1, 2$ to these labels.

Definition 4.2.7 (Trend Prediction) *Given a query keyword,* trend prediction *denotes predicting the label among* Increase*,* Steady *and* Decrease *for the prediction*

window.

Definition 4.2.8 (Problem Definition) *Given an observation period with length l and a corresponding paper collection $< P_1, ..P_l >$, and a query keyword q, the challenged problem is trend prediction of q for a prediction period of length k.*

4.3 Proposed Methods

For keyword trend prediction we propose a family of deep neural models. We can group the models in two, according to the way input is structured. In the first one, we group the paper collection per year and feed it into the model as a long sequence. We call these models as *word embedding based models*. The second group of models take each paper as a separate input. These models are called as *paper embedding based models*. We devise three models within each group. In the word embedding based models, three different architectures are constructed to explore alternative was to handle the long sequence of tokens. In the second group of models, we explore the effect of three different alternatives to obtain paper embeddings on the prediction performance.

Below is a list of components used in the proposed neural models. The architectures include well-known modules, however they are combined in novel ways to be handle the challenged problem.

Embedding: Embedding block maps a word or a paper to a vector.
LSTM: LSTM block is a recurrent module that consumes an array of vector and outputs resulting vector.
Dense: Dense block connects each input to each output, forming densely connected layer.
Lambda: Lambda block applies simple functions such as mapping and reducing tensor dimensions. We basically use this module to be able to map to token stream of a given year.
Reshape: Reshape block changes tensor dimensions, mostly to provide compatibility before concatenation.

Concatenate: Concatenate block appends tensors together in specified dimension.
RepeatVector: RepeatVector block appends the same tensor to specified dimension for a specified amount.
Conv1D: Conv1D is used to apply convolution on temporal dimension.
Dropout: Dropout layer randomly sets input units to 0 with a specified frequency.
GlobalMaxPooling1D: GlobalMaxPooling1D downsamples the input by taking the maximum value over the time dimension.
TimeDistributed: TimeDistributed module allows to apply a layer to every temporal slice of an input.

In the rest of this section, we firstly present how the input is structured for the proposed neural models. Then we describe the proposed neural architectures within each group.

4.3.1 Data Encoding for Neural Models

In the proposed deep neural models, the basic idea is that the model scans all the papers in the observation window. Each paper in a given $year$ is encoded as a series of tokens obtained from the title and the abstract of the paper. Order of the papers is arbitrary as long as they belong to the given year. A paper with index i in year $year$ is represented as given in Equation 4.1, with *n* title tokens and *k* abstract tokens.

$$\begin{aligned} Paper_{year,i} = title_{year,i,1}, ..., title_{year,i,n}, \\ abstract_{year,i,1}, ..., abstract_{year,i,k} \end{aligned} \tag{4.1}$$

Depending on the model, either embedding of the tokens or the whole document is constructed and used. the papers are stacked yearly as in Equation 4.2 and zeros are padded to $maxlen$ of the longest $TokenStream$.

$$TokenStream_{year} = [paper_{year,1}, ..., paper_{year,l}] \tag{4.2}$$

In the experiments, we set the observation window size as 5 due to the conducted validation analysis and the number of years available in the data set. Therefore, the

observation window is represented accordingly in Equation 4.3. However it can be adjusted for any size of observation window.

$$YearlyStream_{year} = \begin{bmatrix} TokenStream_{year-4} \\ TokenStream_{year-3} \\ TokenStream_{year-2} \\ TokenStream_{year-1} \\ TokenStream_{year} \end{bmatrix}_{5 \times maxlen} \tag{4.3}$$

For a query token on a given $year$, data samples used for training can be defined as in Equation 4.4. Query is a single token keyword under consideration for prediction. $label$ belongs to set {-1, 0 , 1} for regression task or set {0, 1, 2} for classification task.

$$x, y = [Year - Stream_{year}, query], label \tag{4.4}$$

4.3.2 Word Embedding Based Neural Models

Models in this group process all the papers of a year in a concatenated form. Each year's paper tokens, which are terms cleaned up from title and abstract, are concatenated, and zero padding is applied to match shorter streams. In this group, we construct three different models where they generate paper collection summary representations in different ways.

Model 1: Year Summary based Model. In this model, the input is a 2-D token array such that each year has its own stream, and all paper tokens in a year are concatenated. After token embeddings are obtained, Model 1 uses a shared *Yearly Summary* module to process one-year stream to construct a year summary vector (shown in Figure 4.2[2]). As shown in the figure, the size of the input 2-D array to the models is *5 years x 60919 tokens per year*[3]. The embedding layer generates embeddings of

[2] Model figures are generated via Netron application [60].
[3] Note that the ? mark in the main input size in Figure 4.2 denotes the number of input batches

vector size 50 for 21742 tokens. The LSTM module processes each of the year based token sequences and generates year summary representations of size 32. The vectors generated for each year in the observation window (5 years) are concatenated and fed into the dense layers to generate the label prediction of the query keyword for the prediction window. Reshape operation aligns query embedding result to LSTM results for concatenation.

Model 2: Model with Observation Window Summary. Model 2 also takes a 2-D token array (5 years x 60919 tokens) as input and uses a shared *Yearly Summary* module to summarize collection of papers per year into a yearly summary vectors of size 32 (shown in Figure 4.3). Differently from Model 1, this model then uses another LSTM module to generate 5-year summary vectors. Input to the second LSTM module is a vector obtained by concatenation of the yearly summary vectors (in the first *Concatenate* module in the figure) and it is further concatenated with the embedding of the query keyword (RepeatVector module and the second *Concatenate* module). The generated summary representation for the observation window (5-year summary representation) of size 32 is then fed into Dense layers for the final output.

Model 3: Convolution based Year Summary Model. As in the previous models, Model 3 takes a 2-D token array as input. This model differs from Model 2 such that it uses shared *convolution* layer to generate yearly paper embeddings (given in Figure 4.4). The convolution layer applies double 1D-convolution with kernel size 5 and 64 filters. After applying Dense layer, yearly summary vectors of size 114 are obtained. The yearly summary vectors and also the query keyword embedding are concatenated as in Model 2 to be given as input to LSTM layer. The rest of the model is also the same as in Model 2, such that summary vector of size 32 is constructed as the representation of the paper collection in the observation period, and then it is fed into the Dense layer for the final output.

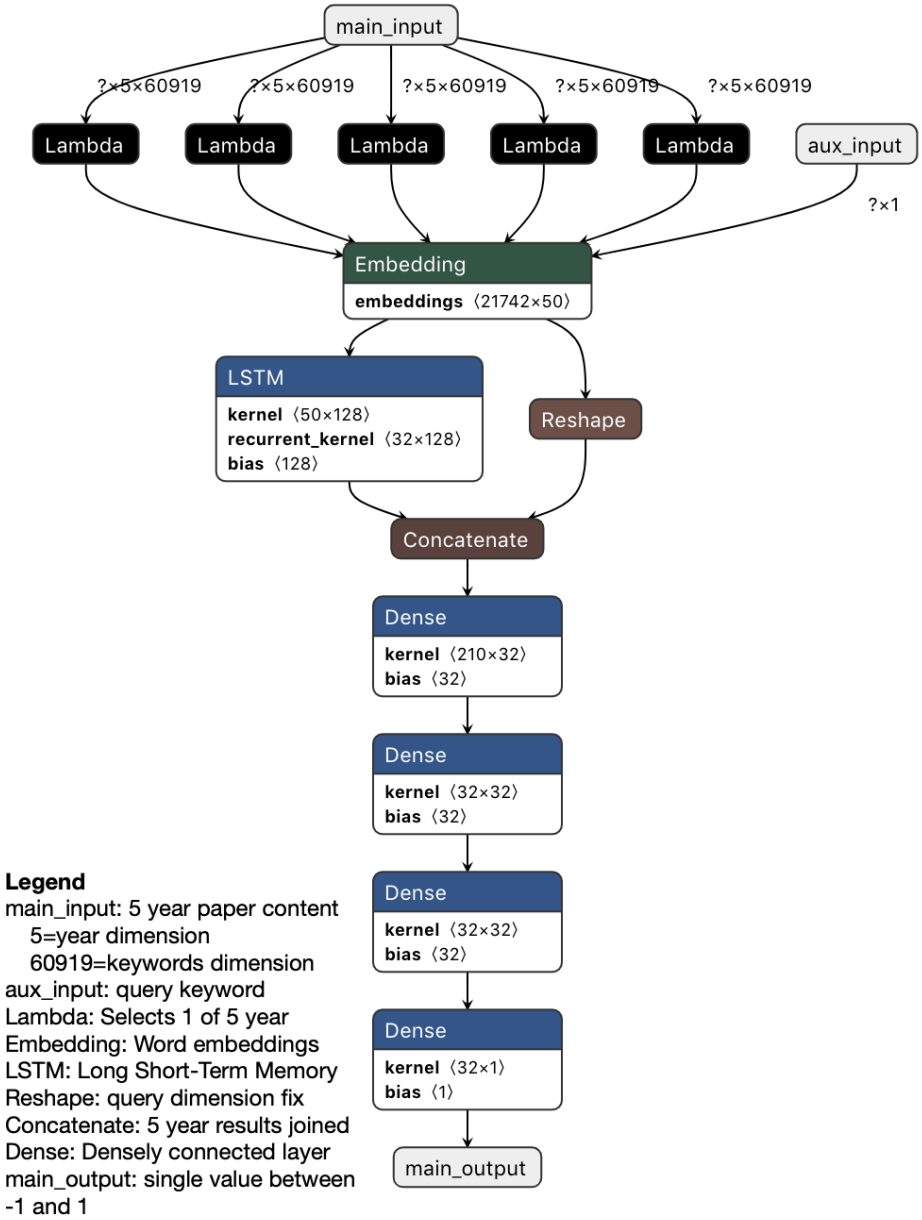


Figure 4.2: Model 1: The architecture includes a shared LSTM module to process paper collections per year, yearly embeddings are concatenated

4.3.3 Paper Embedding Based Models

The paper embedding based models consider a paper as a whole and process each paper individually. Figure 4.5 shows the stages used in these models. Paper vectorization phase takes word embeddings of a paper and produces a vector for the paper. For

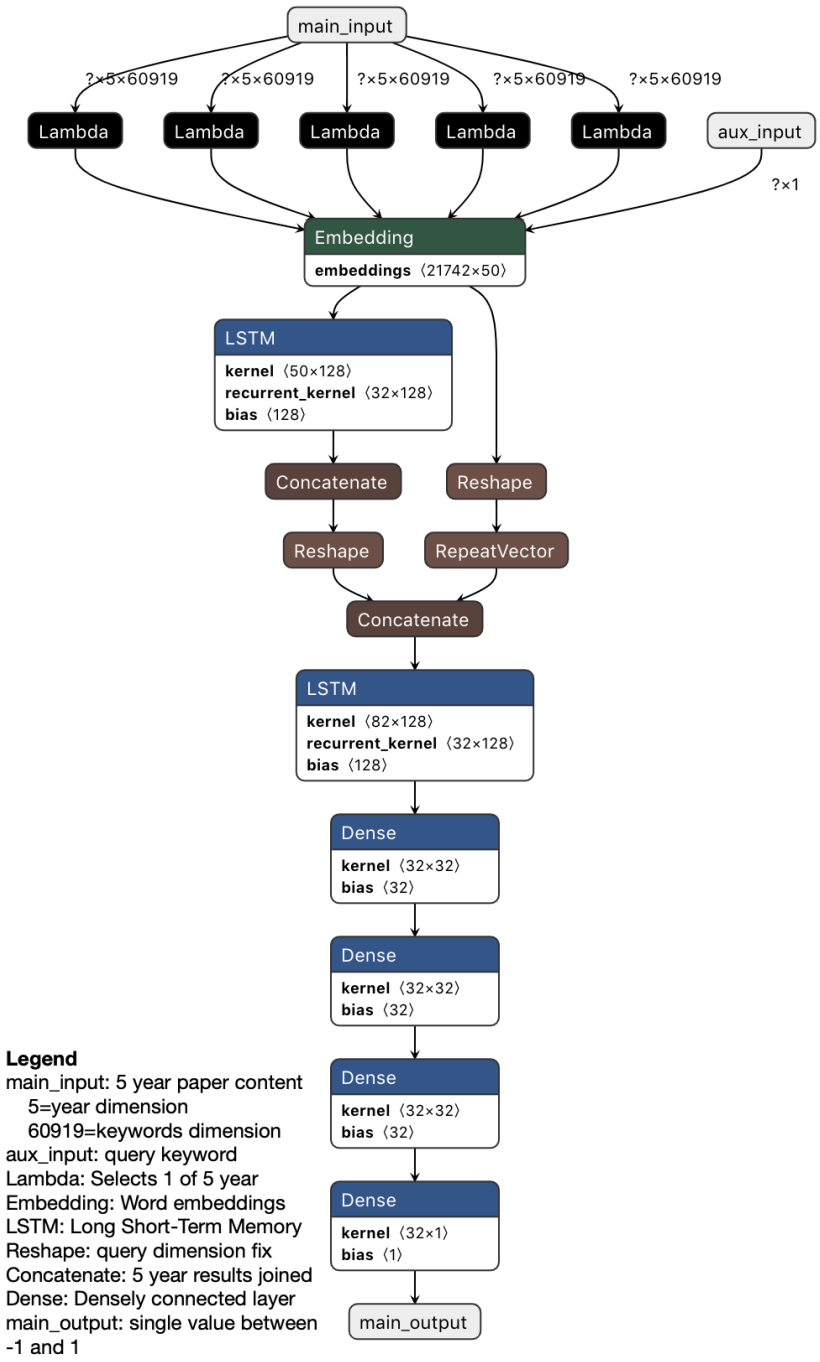


Figure 4.3: Model 2: The architecture includes a shared LSTM module for each year, it includes another LSTM over all year results

paper embedding construction, we use three alternative methods: pre-trained LSTM (*Paper Embedder LSTM*), doc2vec [30], and Specter [31]. After generating a paper vector, LSTM module is utilized to generate the yearly summary vector. As the last

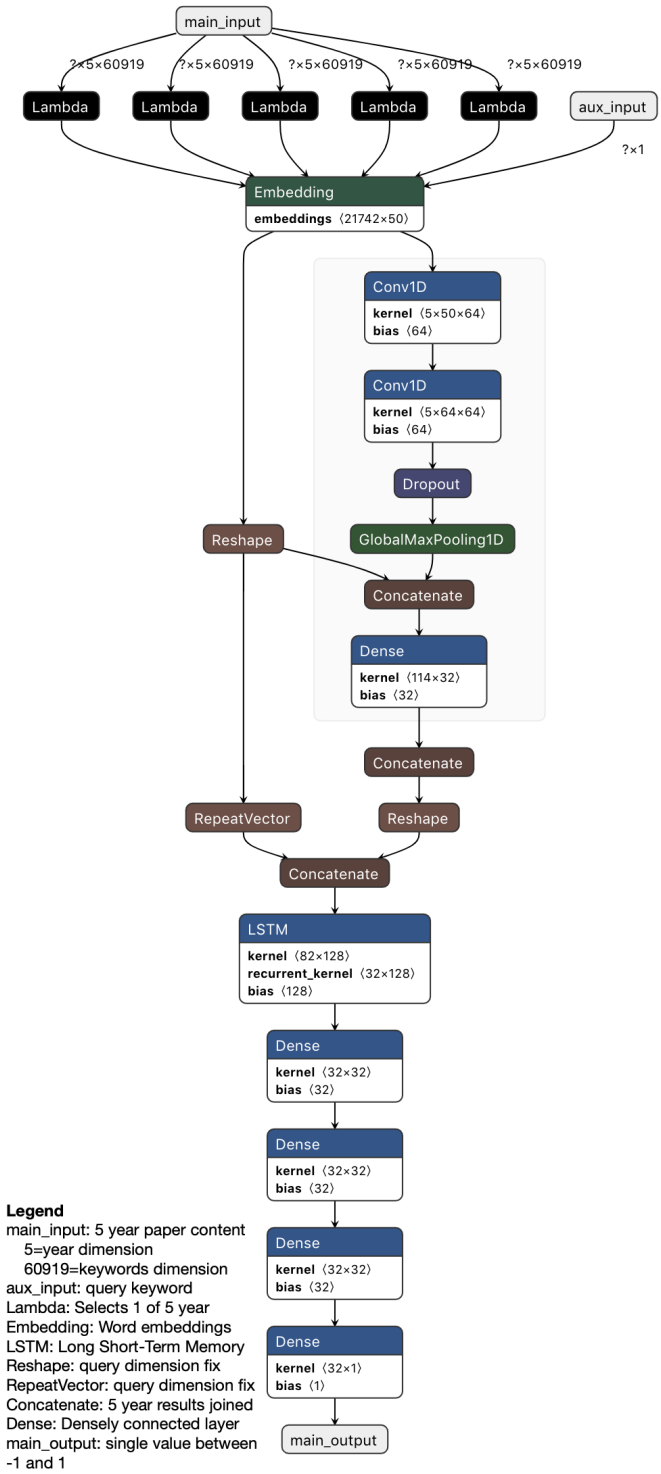


Figure 4.4: Model 3: The architecture includes a Convolution module for each year, it has an LSTM module to process all year results

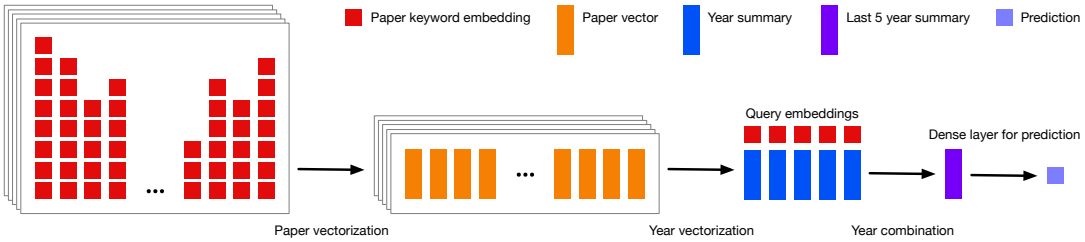


Figure 4.5: Data flow for processing papers to a vector first

stage, another LSTM module is utilized to combine 5 yearly summary vectors with repeated queries (as shown in the third step of the data flow in Figure 4.5). Its result is fed into 3-layer densely connected module, and the final prediction is generated.

Paper Embedder LSTM module is trained beforehand to construct document level representations. For the other two paper embedding based models, we use pre-computed doc2vec and Specter vectors. *Doc2vec* [30] generates paragraph vectors through training on word vectors. On the other hand, *Specter* [31] is a deep learning transformer model specialized for scientific papers. It generates stable document vectors, given the same collection, always generating the same vector. To obtain similar stability with *doc2vec*, we used negative sampling.

Model 4: Paper Embedder LSTM based Model. Figure 4.6 illustrates the deep learning architecture of Model 4. It takes a 3-D token array as input such that *year* dimension holds the year information in the observation window, and *paper* dimension is made up of tokens of each paper, resulting with 3-D array of *5 years of observation window x 410 papers per batch x 385 tokens per paper* in our implementation. This model uses a shared pre-trained *Paper Embedder LSTM* module (*TimeDistributed* module in the figure) to convert all paper tokens into their paper embeddings of size 100. (Note that the TimeDistributed module includes an additional input of embedding collections of size *16350 tokens x embedding size 50*.) All yearly paper vectors are fed into a shared *Year Summary* module (the first LSTM module in the figure) to compute the representation of one-year summary of papers. This module generates a vector of size 50. The input to the second LSTM module is the vector obtained by concatenation of yearly summary vectors and the embedding of the query key-

word. The second LSTM module is used for generating summary representation for the collection within the observation window (5-year summary representation). The resulting embedding is fed into dense layer to generate the trend prediction of the query keyword.

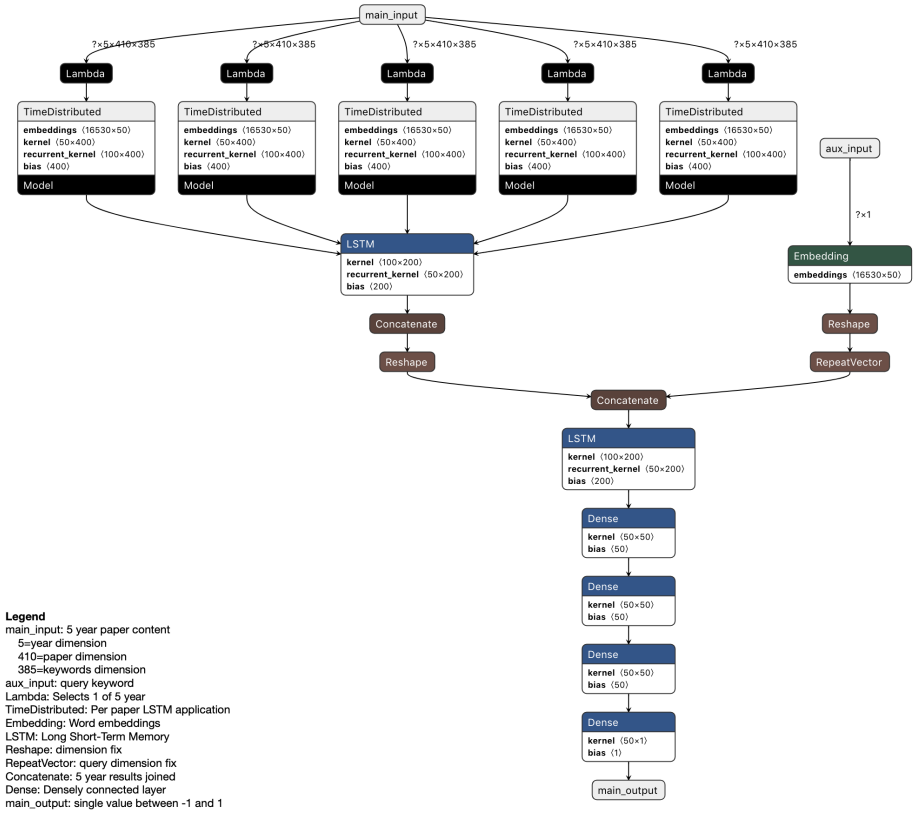


Figure 4.6: Model 4: The model includes *Paper LSTM module* to generate paper embeddings, LSTM module to generate one-year summary embeddings, and another LSTM module to generate representation over the observation window.

Model 5: Doc2vec based Model. Model 5 takes a 3-D array as input such that *year* dimension denotes the size of the observation window, *paper* dimension denotes the size of papers in the batch and *embedding* dimension denotes the size of the doc2vec embedding per paper, resulting with the array of *5 years x 410 papers x 50* in our implementation. The paper vector is already trained with doc2vec implementation on paper tokens that uses global pre-trained word embeddings. Therefore it is similar to Model 4 except that it bypasses paper summary calculations by directly using a

pre-computed doc2vec vector for each paper. Figure 4.7 shows the model of the experiment.

Model 6: Specter based Model. Model 6 also takes a 3-D array as the input. It is similar to Model 5 except that it uses Specter for constructing the paper vectors. Figure 4.7 shows the model.

4.4 Experiments

In this section, we firstly describe the data set, data preparation steps, implementation setup and baseline methods. Following this, we present the keyword trend prediction experiments applied on ten data collections both for all labels and label-wise analysis. Then validation analysis for paper embedding approaches used in the models is presented. Lastly, we present the qualitative analysis conducted for trend prediction on three of the venues.

4.4.1 Data Set and Experiment Setup

As the data set, we use a collection of papers obtained from Microsoft Academic Graph [61]. It is an academic collaboration network data set containing papers with year, title, abstract, list of authors, list of citations, and venue information. Additionally, Microsoft Academic Graph associates each paper with the related fields of study. Field of study is a hierarchic graph that includes topics and concepts. In our analysis, we use year, title, abstract and venue information of the papers, and we consider the fields of study as the keywords of a paper. Unfortunately Microsoft Academic Graph product was shutdown at the end of 2021 which increased difficulty in data gathering for academic research.

For the experiments, ten computer science venues and their papers published between 2001 and 2013 are selected. We consider that each venue has some differences in focused themes and aim to discover trends within each venue. The venues are listed

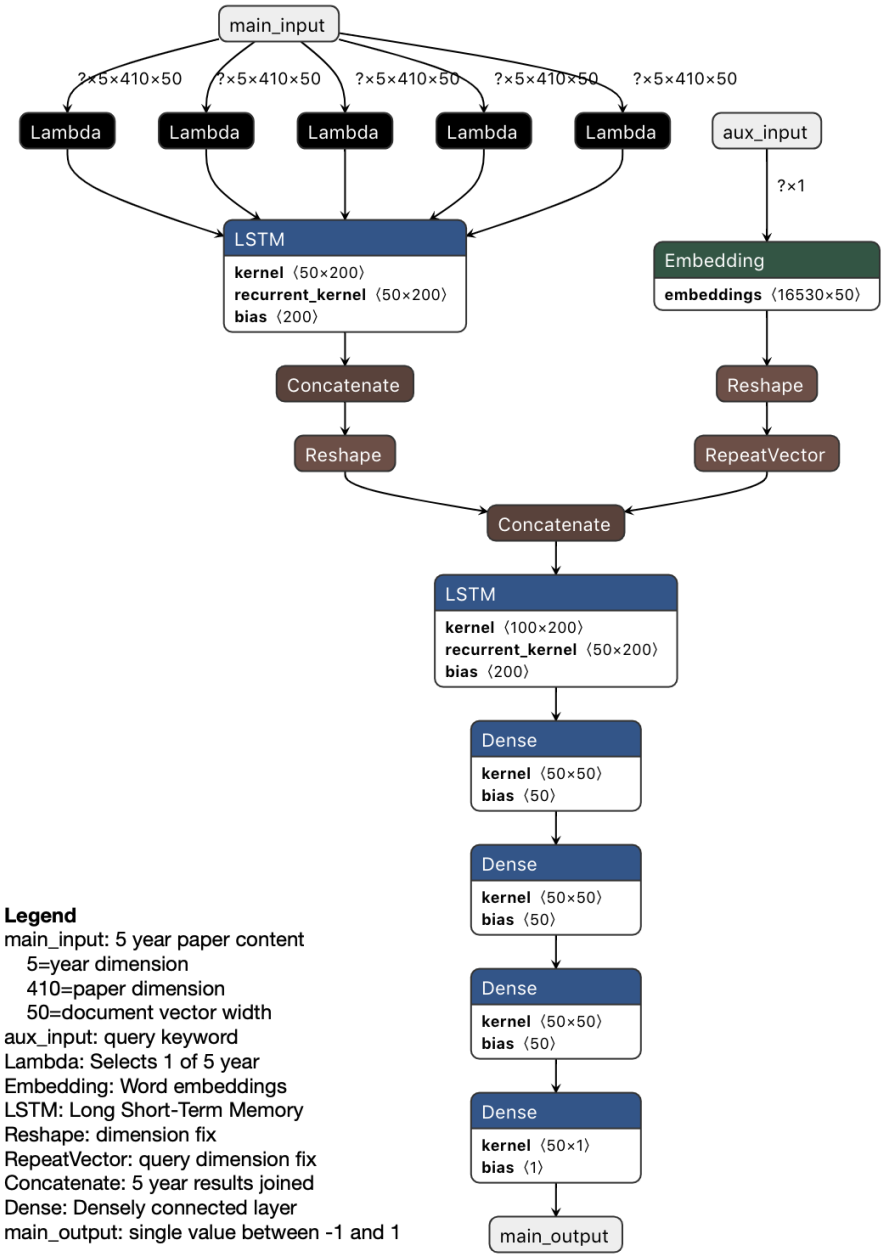


Figure 4.7: Model 5 & 6: The model includes Doc2Vec (in Model 5) or Specter (in Model 6) for each paper, an LSTM module to generate one-year summary representation, an another LSTM module to generate representation over the observation window.

in alphabetical order in Table 4.1. The table also lists the number of papers for each venue, and the average number of tokens in the title and abstract of the papers.

Each entry in the data collection is a paper instance with year and keywords. Paper title and abstract are considered as a whole text. Keywords are lemmatized by using spaCy Python package[4]. Stopwords are also removed from keyword set.

To be used as query keywords in the training, validation and testing phases, top 500 keywords are chosen according to the frequency. For each year, we keep a balanced keyword collection in terms of labels, and the number of instances per year is determined as at most 102 for manageable experiment run-time.

The data set for each venue is partitioned such that 70% is used in training, 15% is used for development in training and hyper-tuning phase, and the remaining 15% is used solely in testing. As an example, Figure 4.8 illustrates how the paper collection between the years 2001 and 2013 are partitioned. Here, the blue squares depict the usage in the observation window and the orange ones depict the usage in the prediction window. The label for a query keyword is determined as given in Definition 4.2.6. Training data set contains 4 streams of data. For example, in the first stream (train stream 1), labels of the training query keywords for the prediction window (window of 2006-2008) is determined by using the paper collection in the observation window (window of 2001-2005). As described in Definition 4.2.6, a single label is generated for the prediction window per query keyword. The fifth stream is used as the validation data set. Test data stream uses the model constructed from the earlier streams and the predictions of the test query keywords are generated for the prediction window of 2011-2013.

4.4.2 Implementation Setup

For the word embedding based models, we use pre-trained word embedding weights from a ready source, GLoVe [53]. GloVe provides 400,000 words trained and has different embedding sizes, which are 25, 50, 100, 200 and 300. In our experiments, GLoVe 50 is used and we have observed that further training for fine tuning does not change their weights. For doc2vec implementation, Gensim library [62] is used. For Specter paper embedding model, implementation provided on Github [5] is used.

[4] https://spacy.io/
[5] https://github.com/allenai/specter

Table 4.1: Selected venues in the data set

Abbr.	Venue name	Paper count	Avg. tokens
AAAI	Association for the Advancement of Artificial Intelligence	6430	83
CIKM	The Conference on Information and Knowledge Management	3457	98
DSS	Decision Support Systems	2166	90
ICDE	International Conference on Data Engineering	2888	92
ICDM	International Conference on Data Mining	3447	91
KDD	Knowledge Discovery and Data Mining	3555	100
SIGIR	International ACM SIGIR Conference on Research and Development in Information Retrieval	2797	87
SIGMOD	International Conference on Management of Data	2553	93
VLDB	Very Large Data Bases	2828	102
WWW	The Web Conference	3963	84

The proposed deep learning architectures, LSTM and CNN modules are developed by using the Keras library [63].

4.4.3 Baseline Models

We adopted four baseline methods, two regression based and two classification based ones. Linear regression and Support Vector Regression are used as the regression based baseline methods. For classification based baseline methods, Logistic regression (with class labels) and Support Vector Classification are utilized. We selected the baseline methods in order to see whether the challenged problem can be handled by basic supervised learning approaches. Although there are several related studies as summarized in Section 2.1, there are incompatibilities in problem definition and data modeling, they are not directly comparable and hence not suitable as baseline. All baseline methods are implemented by using scikit-learn [64] package.

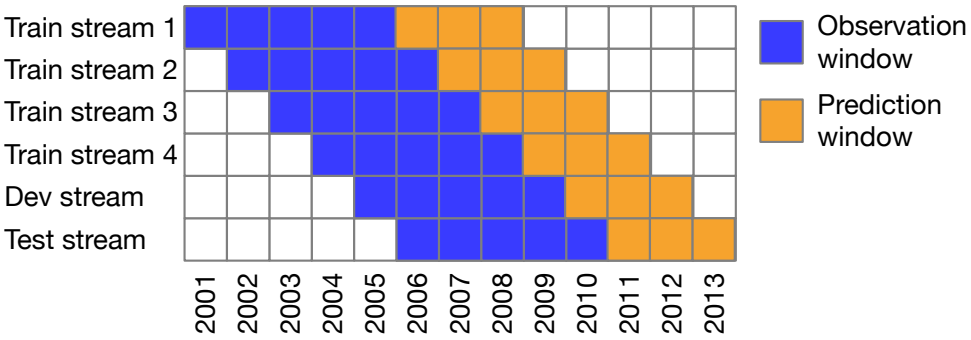


Figure 4.8: Illustration of data set splits and time windows

Baseline methods use the count of query tokens per year as their single feature. Therefore, for an observation window of length 5, a sequence of 5 values constitutes the input. This is formalized in Equation 4.5 in the form of a simple matrix. $TokenStream_t$ denotes a single array of all paper tokens belonging to year t. For each training and test sample, corresponding $BaselineYearlyStream$ is calculated.

$$BaselineYearlyStream(query, year) = \begin{bmatrix} count(query, TokenStream_{year-4}) \\ count(query, TokenStream_{year-3}) \\ count(query, TokenStream_{year-2}) \\ count(query, TokenStream_{year-1}) \\ count(query, TokenStream_{year}) \end{bmatrix}_{5\times1} \tag{4.5}$$

4.4.4 Keyword Trend Prediction Analysis

In the experiments, we evaluate the performance of the proposed models and the baselines in terms of precision, recall, and F1-score. For F1-score, we follow macro averaged setting as it gives equal weights to each class [65].

The methods in the result tables are grouped into three: the upper group contains the baseline models, i.e., LinReg, SVR, LogReg, and SVC. The middle group contains word embedding based models, Model 1, Model2, and Model 3, while the lower group contains the paper embedding based models, Model 4, Model 5, and Model 6.

Table 4.2: Precision results of the models

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	NaN	0.46	NaN	NaN	NaN	NaN	NaN	0.78	NaN	0.70	NaN	NaN
SVR	NaN	0.34	NaN	NaN	NaN	NaN	NaN	0.44	NaN	0.56	NaN	NaN
LogReg	0.36	NaN	NaN	NaN	NaN	0.31	NaN	0.36	0.50	0.46	NaN	NaN
SVC	0.34	0.43	0.44	NaN	0.59	0.37	0.28	0.32	0.39	0.36	NaN	0.34
Model1	0.32	0.38	0.43	0.38	0.47	0.39	0.39	0.40	0.38	0.37	0.39	0.38
Model2	0.33	0.44	0.37	0.40	0.45	0.32	0.48	0.47	0.39	0.39	0.40	0.54
Model3	0.27	0.38	0.38	0.33	0.46	0.38	0.52	0.43	0.37	0.47	0.40	0.43
Model4	0.29	0.44	0.39	0.36	0.46	0.36	0.46	0.38	0.38	0.36	0.39	0.45
Model5	0.29	0.41	0.40	0.37	0.47	0.41	0.48	0.42	0.40	0.44	0.41	0.53
Model6	0.29	0.52	0.39	0.37	0.53	0.39	0.51	0.51	0.35	0.39	0.42	0.35

Table 4.3: Recall results of the models

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	0.33	0.37	0.37	0.34	0.36	0.37	0.40	0.36	0.34	0.36	0.36	0.40
SVR	0.32	0.32	0.38	0.33	0.37	0.37	0.41	0.34	0.36	0.36	0.36	0.35
LogReg	0.37	0.43	0.35	0.33	0.34	0.34	0.43	0.37	0.46	0.44	0.39	0.33
SVC	0.32	0.45	0.38	0.35	0.38	0.37	0.38	0.33	0.38	0.35	0.37	0.36
Model1	0.34	0.36	0.41	0.40	0.46	0.39	0.39	0.37	0.37	0.36	0.39	0.35
Model2	0.33	0.41	0.36	0.39	0.45	0.32	0.47	0.47	0.39	0.40	0.40	0.52
Model3	0.29	0.36	0.38	0.32	0.42	0.37	0.45	0.37	0.37	0.41	0.37	0.42
Model4	0.31	0.41	0.35	0.34	0.46	0.35	0.46	0.36	0.38	0.35	0.38	0.45
Model5	0.29	0.40	0.38	0.35	0.45	0.41	0.49	0.41	0.39	0.41	0.40	0.49
Model6	0.30	0.45	0.35	0.33	0.45	0.39	0.49	0.45	0.33	0.35	0.39	0.41

For better readability, we colored the cells with top-3 results per venue, such that the dark green cells indicate the best results, while the green and light green cells indicate, respectively, the second best and third best results for a given data set. The NaN values are observed for precision of baseline models. Cells with this value indicate that the model has made no prediction for a label. The last column of the tables reports the average metric results per method is reported in the column *Venue Average*.

In order to evaluate the trend prediction for a broader research area, we conducted an additional analysis by combining the paper collections of all the venues. The last column (named as *All*) in the tables include the result of this analysis.

In Table 4.2, the precision results are given. The results indicate that the baseline models achieve the highest scores for 6 venues out of 10. However, vast number of NaaN values for baseline models show that baseline models fail to learn some of the labels. Word embedding models score the best precision results for two venues, which is also the case for paper embedding based models. When the averaged precision values are examined, paper embedding models (Model 6 and Model 5) score the highest values followed by word embedding based model (Model 2 and Model 3).

For the precision under *ALL* venues, the gap between the prediction performance of the proposed models and the baselines becomes more clear. In this analysis, it is seen that the increase in the amount of evidence obtained in the observation window positively affects the performance of the proposed neural models. The performance results in all metrics increase compared to venue based results. In this analysis, Model 2 and Model 5 have the top two scores, respectively.

In Table 4.3, the recall results are reported. The baseline models and the word embedding models achieve the highest score for four academic venues, while the paper embedding based models achieve the highest scores for five of them. Logistic Regression performs better than the other baseline models, as it achieves the highest 3 scores among 4 achieved by baseline models. When word embedding models are compared, Model 1 scores the highest results for 3 venues. In case of paper embedding models, Model 5 and Model 6 have a tie with two highest results. However, for average recall values, the neural models perform better than the baseline models.

Table 4.4: Macro averaged $F1$ score of the models

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	NaN	0.28	NaN	NaN	NaN	NaN	NaN	0.23	NaN	0.24	NaN	NaN
SVR	NaN	0.26	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.27	NaN	NaN
LogReg	0.31	NaN	NaN	NaN	NaN	0.21	NaN	0.36	0.43	0.37	NaN	NaN
SVC	0.22	0.40	0.31	NaN	0.31	0.35	NaN	0.32	0.35	0.34	NaN	0.33
Model1	0.30	0.34	0.37	0.36	0.42	0.38	0.38	0.36	0.36	0.35	0.36	0.34
Model2	0.33	0.41	0.35	0.38	0.44	0.31	0.46	0.46	0.38	0.39	0.39	0.50
Model3	0.26	0.36	0.38	0.27	0.39	0.36	0.43	0.36	0.35	0.40	0.36	0.41
Model4	0.29	0.41	0.35	0.34	0.45	0.35	0.45	0.37	0.37	0.35	0.37	0.44
Model5	0.29	0.40	0.37	0.36	0.44	0.39	0.48	0.41	0.39	0.41	0.39	0.48
Model6	0.27	0.46	0.32	0.31	0.42	0.38	0.48	0.44	0.31	0.33	0.37	NaN

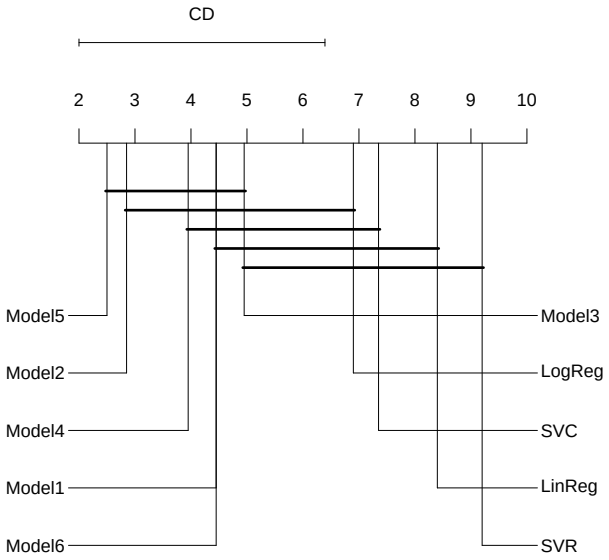


Figure 4.9: Nemenyi post-hoc test result for Model $F1$ macro averaged score

For the recall obtained for *ALL* venues, as in the precision results, we see that the proposed models provide much better performance than the baseline methods. In recall analysis, again, Model 2 and Model 5 have the top two results.

In Table 4.4, we report the macro averaged F1-scores. As the results indicate, the neural models achieve higher F1-scores compared to the baseline models. The baseline models achieve the highest F1-score only for one venue (Logistic Regression for VLDB). The paper embedding models perform better compared to word embedding models. The paper embedding models record 6 top scores compared to 4 achieved by the word embedding models.

For the F-1 scores under *ALL* venues, in parallel to the previous results, the proposed models provide better trend prediction performance with a clear gap over the baseline methods. In this analysis also, Model 2 and Model 5 give the best performance results.

4.4.5 Statistical Significance Analysis of F1-Score Results

As F1-score provides more insights about the results compared to recall and precision, we statistically analyze F1-scores. To this aim, we employe Iman-Davenport test [66] and Nemenyi post-hoc test [67]. Iman-Davenport test is a non-parametric test

to compare performance of multiple algorithms by their variance of ranks. The null hypothesis of the test assumes that the algorithms do not differ. Nemenyi post-hoc test is used to determine the statistically differing algorithms once the null hypothesis of Iman-Davenport is rejected (p-value$<$0.05). In the graphical representation of Nemenyi post-hoc test, the algorithms are sorted by their ranks, and algorithms that do not statistically differ are connected via vertical lines. The CD ruler in the graphical representation indicates the critical difference. Two algorithms statistically differ if the difference of their mean ranks exceeds the critical difference.

The Iman Davenport test returns p-value = 2.045e-12, which indicates that methods statistically differ. Figure 4.9 depicts Nemenyi post-hoc test results. As the figure indicates, the neural models have higher average ranks compared to the baseline models. Model 5 statistically differs from all baseline models. Similarly, Model 2 also statistically differs from all baseline models other than Logistic Regression.

4.4.6 Label-wise Keyword Trend Prediction Performance Analysis

Since predicting the topics having increasing trend has more value in practical use, we further report and analyze the precision, recall, and F1-score values for the label *Increase*. Table 4.5 reports the precision results. As the results indicate, the baseline models fail to learn the *Increase* label for several data sets, as the corresponding result is a NaN. Logistic Regression and SVC perform better than the other baseline models, they fail to learn the *Increase* label only for one data set. The baseline models have the highest precision score for six cases, while the neural models for eight cases. Among the neural models, Model 2 and Model 4 achieve the highest results for two data sets. Model 5 has the highest average recall value.

For the precision under *ALL* venues, the gap between the proposed models and the baselines becomes even more clear emphasizing the advantage of the proposed models. As in the previous experiments on*ALL* venues, the performance results in all metrics increase compared to venue based results. In this analysis, Model 6 and Model 2 have the top two results, respectively.

In Table 4.6, we report the recall results for label *Increase*. The results are analogous

to those observed for the precision. Model 2 achieves the highest average recall result. Among the neural models, Model 2 and Model 4 score two top results, however Model 5 scores within the top three scores for each data set. Logistic Regression performs better than the other baseline models. It achieves the highest recall score for five data sets out of ten.

For the recall obtained for *ALL* venues, as in the precision results, we see that the proposed models provide much higher recall performance than the baseline methods. In this analysis, differently from the precision results, Model 4 has the top rank, whereas Model 5 and Model 6 both have the second-best recall performance for the label *Increase*.

In Table 4.7, the macro averaged F1-scores for the *Increase* label are reported. As seen in the table, the neural models ranks in top 3 for the most of the experiments compared to the baseline models. Model 5 has the highest average F1-score, however it does not achieve the highest score for any of the venues. Model 2 has the second best average F1-score and achieves the highest scores for two of the venues.

For the F1-score obtained for *ALL* venues, as in the previous results, we see that the proposed models show a clear advantage over the baseline methods. In this analysis, Model 6 has the top rank, whereas Model 4 has the second-best F1-score performance for the label *Increase*.

4.4.7 Statistical Significance Analysis of Label-wise Performance Results

We statistically analyze the F1-scores for the label *Increase*, as well. Iman Daveport test gives p-value = 4.448e-11, which indicates that methods statistically differ. Figure4.10 depicts the Nemenyi post-hoc test results. As the result indicate, Model 2 has the highest mean average rank and statistically differ from Linear Regression and SVR. Model 5 has the second best mean average rank and statistically differs from Linear Regression and SVR. Logistic Regression, a baseline model, ranks better than Model 1 and Model 3, however it does not statistically differ from the proposed neural models.

Table 4.5: Precision results of the models for label *Increase*

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	NaN	0.55	NaN	NaN	NaN	NaN	NaN	1.00	0.33	0.75	NaN	NaN
SVR	0.00	0.44	NaN	NaN	NaN	NaN	NaN	1.00	NaN	0.60	NaN	NaN
LogReg	0.47	0.41	0.33	NaN	0.40	0.20	0.44	0.42	0.40	0.45	NaN	NaN
SVC	0.40	0.46	0.25	NaN	1.00	0.38	0.00	0.35	0.36	0.26	NaN	0.14
Model1	0.29	0.34	0.50	0.50	0.52	0.41	0.45	0.47	0.33	0.33	0.41	0.33
Model2	0.26	0.55	0.40	0.48	0.44	0.29	0.48	0.51	0.29	0.46	0.42	0.64
Model3	0.27	0.42	0.45	0.33	0.48	0.39	0.46	0.53	0.30	0.62	0.42	0.39
Model4	0.27	0.62	0.35	0.32	0.44	0.34	0.44	0.42	0.38	0.32	0.39	0.50
Model5	0.29	0.52	0.39	0.48	0.44	0.42	0.49	0.50	0.40	0.52	0.45	0.56
Model6	0.25	0.79	0.30	0.38	0.50	0.37	0.49	0.58	0.25	0.38	0.43	0.69

Table 4.6: Recall results of the models for label *Increase*

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	0.00	0.19	0.00	0.00	0.00	0.00	0.00	0.06	0.03	0.09	0.04	0.00
SVR	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.09	0.03	0.00
LogReg	0.21	0.81	0.04	0.00	0.06	0.01	0.36	0.62	0.71	0.62	0.34	0.00
SVC	0.06	0.66	0.08	0.00	0.03	0.15	0.00	0.56	0.12	0.18	0.18	0.09
Model1	0.12	0.34	0.12	0.52	0.44	0.26	0.45	0.25	0.32	0.18	0.30	0.30
Model2	0.21	0.38	0.33	0.29	0.52	0.27	0.48	0.56	0.24	0.55	0.38	0.32
Model3	0.18	0.25	0.42	0.03	0.39	0.24	0.52	0.24	0.18	0.29	0.27	0.30
Model4	0.21	0.47	0.33	0.24	0.42	0.31	0.45	0.32	0.26	0.21	0.32	0.48
Model5	0.18	0.47	0.29	0.41	0.42	0.24	0.55	0.47	0.29	0.38	0.37	0.39
Model6	0.15	0.47	0.12	0.15	0.39	0.26	0.61	0.32	0.12	0.18	0.28	0.39

Table 4.7: $F1$ score results of the models for label *Increase*

	AAAI	CIKM	DSS	ICDE	ICDM	KDD	SIGIR	SIGMOD	VLDB	WWW	Venue Avg	ALL
LinReg	NaN	0.28	NaN	NaN	NaN	NaN	NaN	0.11	0.05	0.16	NaN	NaN
SVR	NaN	0.20	NaN	NaN	NaN	NaN	NaN	0.11	NaN	0.15	NaN	NaN
LogReg	0.29	0.55	0.07	NaN	0.11	0.03	0.40	0.50	0.51	0.52	NaN	NaN
SVC	0.10	0.54	0.12	NaN	0.06	0.21	NaN	0.43	0.18	0.21	NaN	0.11
Model1	0.17	0.34	0.20	0.51	0.47	0.32	0.45	0.33	0.33	0.24	0.34	0.32
Model2	0.23	0.44	0.36	0.36	0.47	0.28	0.48	0.54	0.26	0.50	0.39	0.42
Model3	0.21	0.31	0.43	0.05	0.43	0.29	0.49	0.33	0.22	0.40	0.32	0.34
Model4	0.23	0.54	0.34	0.27	0.43	0.33	0.45	0.37	0.31	0.25	0.35	0.49
Model5	0.22	0.49	0.33	0.44	0.43	0.31	0.51	0.48	0.34	0.44	0.40	0.46
Model6	0.19	0.59	0.18	0.21	0.44	0.31	0.54	0.42	0.16	0.24	0.33	0.50

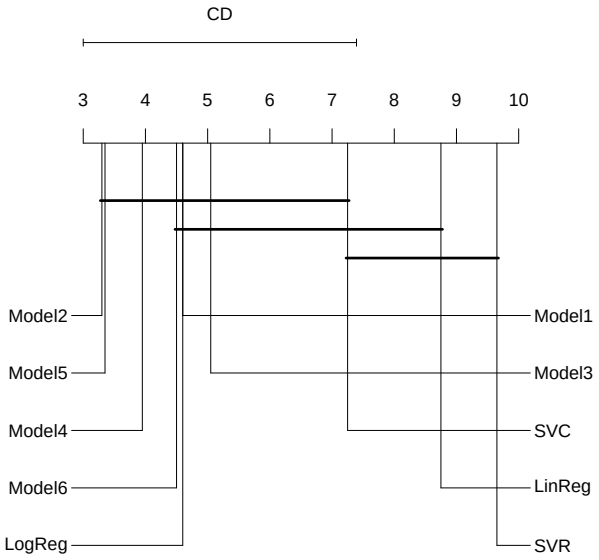


Figure 4.10: Nemenyi post-hoc test result for Label *Increase* $F1$ score

4.4.8 Validation Analysis for Paper Embedding

In order to validate the use of paper embedding approaches, we present an addition analysis on a basic classification task by using the *Paper Embedder LSTM*, which is used as a pre-trained LSTM in Model 4, *doc2vec* of Model 5 and the *Specter* of Model 6. The pre-training task uses all of the papers in the data collection. Each paper has an associated field of study, represented as a one-hot vector for the top 100 field of studies. The data set is divided into training, development and test, with the ratios of 70%, 15%, and 15%, respectively. A multi-label classification task to predict the field of the paper is applied. In the experiment, *PaperEmbedderLSTM* is trained on all papers of the venues without any parameter learning.

As seen in Table 4.8, all paper embedding models provide satisfactory results for the classification task, and the result of the *Paper Embedder LSTM* is comparable with the other embedding models.

Table 4.8: Field of Study Classification results on paper embeddings

Experiment	Accuracy
$PaperEmbedderLSTM$	0.96937
doc2vec embeddings	0.96940
Specter embeddings	0.97056

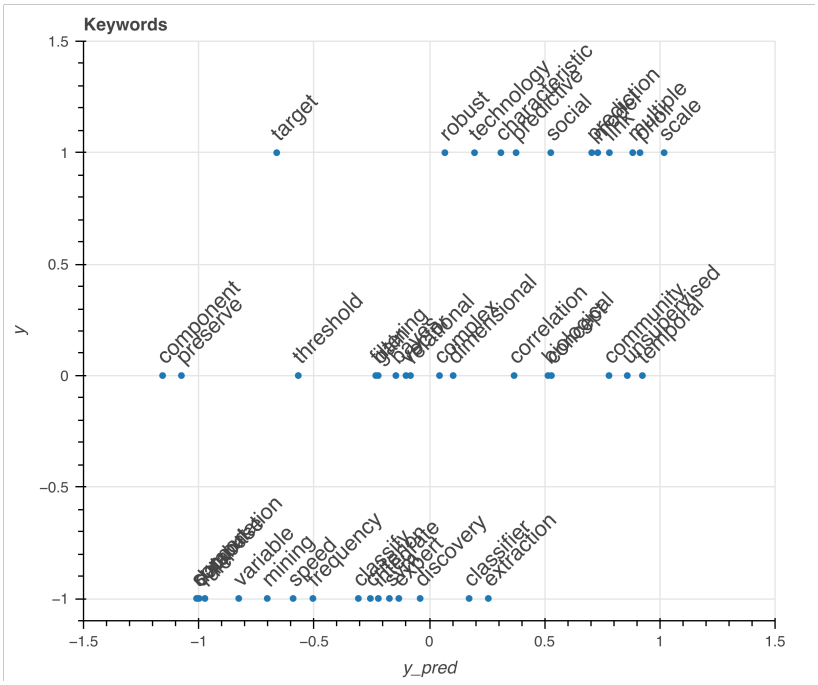


Figure 4.11: Keyword plot for the ICDM

4.4.9 Qualitative Analysis

In order to present a qualitative analysis of the proposed method, we obtain the trend predictions for a set of keywords for publications of the ICDM, VLDB and WWW conferences by using Model 4. For all three conferences, the model is trained by the publications in 2001 and 2013. The trend prediction is performed for the year 2010. For each conference we sort the most frequent terms used and manually select the top 50 relevant terms for the conference. Hence although there are overlaps, each conference has its own set of keywords. The predictions are plotted against the real trend values as given in Figure 4.11, Figure 4.12 and Figure 4.13 for ICDM, VLDB and WWW, respectively.

Table 4.9: Correctly predicted keywords for the ICDM

Decrease	Steady	Increase
computation	bayes	link
database	complex	model
frequency	correlation	multiple
mining	dimensional	prediction
rule	filtering	prior
speed	gain	scale
support	relational	social
variable	vector	

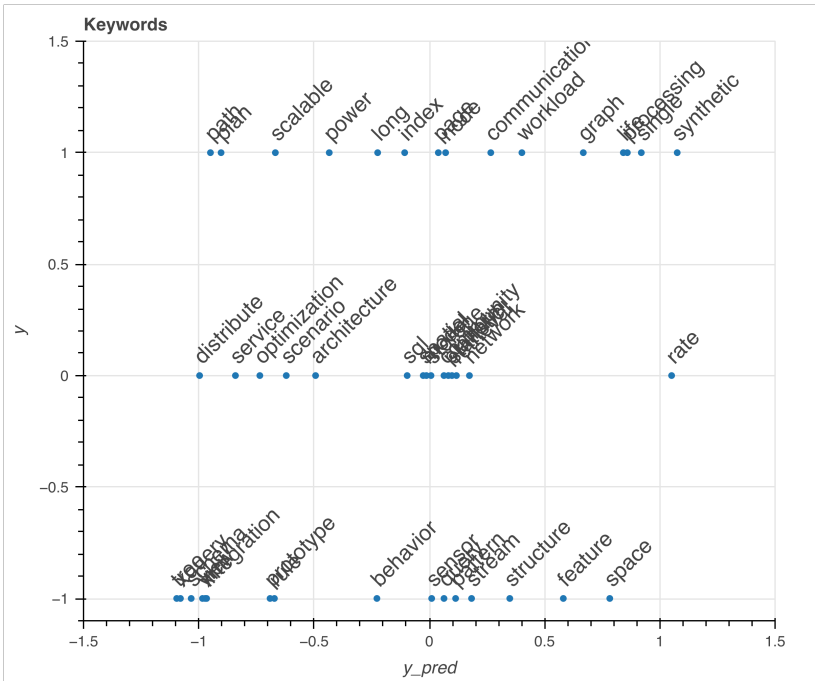


Figure 4.12: Keyword plot for the VLDB

In general, the keywords along the diagonal of the charts are those whose trends are correctly predicted. In the figures, the region around the diagonal includes a high number of keywords denoting the prediction success for all three venues.

For the ICDM conference, the correct prediction for the increase in the use of the term *scale* is inline with the increasing efforts on big data. Similarly the correctly predicted

Table 4.10: Correctly predicted keywords for the VLDB

Decrease	Steady	Increase
integration	architecture	graph
prototype	community	life
rule	model	processing
schema	network	single
tree	platform	synthetic
view	retrieval	
xml	spatial	
xquery	sql	
	statistic	
	storage	

trend for the term *prediction* in ICDM conference is in parallel with the increase in analytics focused studies in the conference. An interesting observation is that that there is a decrease in the trend of the keywords *computational* and *speed*, which are correctly predicted. Although these keywords are closely related with the keyword *scale* (scalability), the proposed method can determine the change in the language for expressing similar concepts. Correctly predicted keywords for the venue ICDM are given in Table 4.9.

For the VLDB conference, the trend prediction for the keywords *synthetic* appears to be compatible with analytics focused studies and the increase in the use of synthetic data sets. Although there is a slight mismatch between the predicted and the real trend values, the keyword *graph* reflects the increase in the use of graph based modelling and processing. On the other hand, the correctly predicted decrease for the terms *integration* and *schema* shows the declining number of studies on such comparatively classical and mature topics. Correctly predicted keywords of the venue are given in Table 4.10.

The figure for the WWW conference shows a correct trend prediction for the term *scale*, which is also a trending keyword for the ICDM conference. Additionally, the predictions for the keywords *social, network* and *community* are inline with the

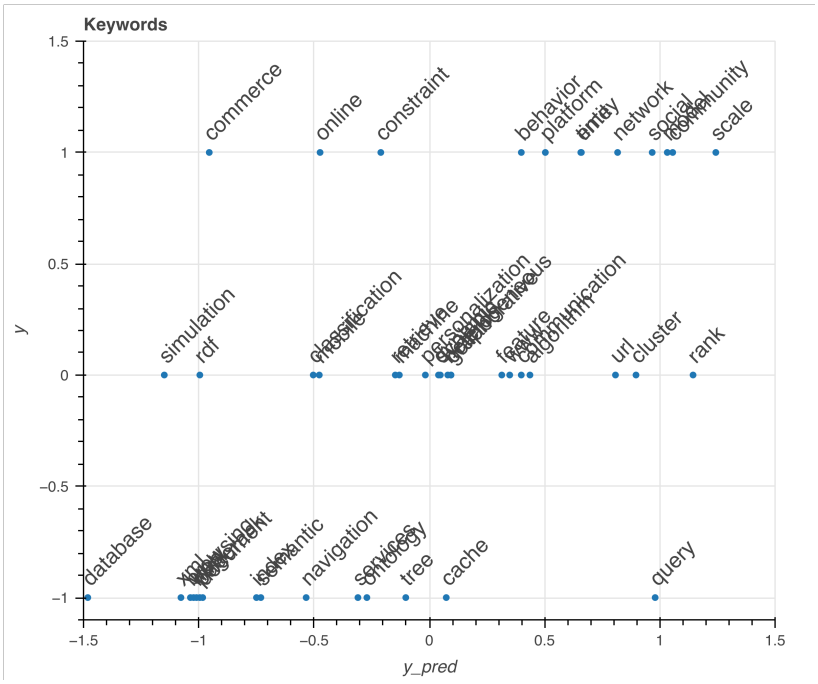


Figure 4.13: Keyword plot for the WWW

Table 4.11: Correctly predicted keywords for the WWW

Decrease	Steady	Increase
browsing	algorithm	community
database	collaborative	entity
document	communication	model
html	dynamic	network
index	feature	platform
navigation	graph	scale
pagerank	heterogeneous	social
semantic	machine	time
web	mobile	
xml	personalization	
	retrieve	
	scalable	
	www	

Table 4.12: Confusion matrices of the word embedding based models

		ICDM predictions			VLDB predictions			WWW predictions		
	Label	-1	0	1	-1	0	1	-1	0	1
Model1	-1	5	21	8	8	14	12	14	16	4
actual	0	2	27	5	5	19	10	9	17	8
labels	1	4	14	14	5	18	11	8	19	6
Model2	-1	9	13	12	13	9	12	10	16	8
actual	0	5	19	10	7	19	8	9	12	13
labels	1	5	11	17	8	18	8	7	8	18
Model3	-1	6	21	7	12	14	8	11	22	1
actual	0	4	23	7	8	20	6	8	21	5
labels	1	1	19	13	8	20	6	7	17	10

Table 4.13: Confusion matrices of the paper embedding based models

		ICDM predictions			VLDB predictions			WWW predictions		
	Label	-1	0	1	-1	0	1	-1	0	1
Model4	-1	14	10	10	10	16	8	15	14	5
actual	0	8	18	8	7	20	7	10	14	10
labels	1	4	15	14	8	17	9	6	21	7
Model5	-1	11	13	10	13	13	8	12	19	3
actual	0	6	20	8	10	17	7	8	17	9
labels	1	3	16	14	8	16	10	6	15	13
Model6	-1	7	20	7	9	21	4	9	22	3
actual	0	3	25	6	5	21	8	8	21	5
labels	1	0	20	13	4	26	4	7	17	10

increase in social network analysis related studies under WWW conference. Similar to the analysis for VLDB, we observe a correctly predicted for the use of the terms *xml, semantic* and *index*, which have a certain level of maturity. Correctly predicted keywords are in Table 4.11.

The table 4.12 shows the confusion matrix of the word embedding based models

whereas the table 4.13 shows for the paper embedding based models. For these regression based models, the largest loss comes from false prediction of farthest labels that are -1 (which represents label *Decrease*) and 1 (which represents label *Increase*). It can be observed from the confusion matrices that these values are generally small. Another observation is that the models can utilize each label for prediction but the errors show they have a slight prediction tendency for 0 which is label *Steady*.

4.5 Discussion

In this chapter, we study the problem of predicting the trends in academic topics. Main motivation of the study is to explore the use of deep neural architectures without using hand-crafted features and additional information other than paper collections. Instead of keyword frequency based approach, we keep track of the trends in terms of change in the frequency distributions of keywords. Given a query keyword, we define three labels, *Increase*, *Decrease* and *Steady*, denoting increase, decrease or no change in the frequency distribution for the prediction time window. The proposed solution scans the academic papers as a token sequence per year. We propose a family of deep neural architectures that process the sequence of tokens for each year in the observation window. Due to this sequence based nature of the problem, LSTM module has a core position in the proposed architectures, however it is combined with other modules in a novel setting. In all the proposed models, generating representation/embedding for year based summary of the paper collections and observation window based summary of the collections have a crucial role. We can group the proposed architectures in two, word embedding based and paper embedding based models. In the first one, the proposed architectures differ from each other as to how these summaries are constructed. For the second group, we explore the use of three different document embedding approaches.

For the experiments, we use Microsoft Academic Network data set and conduct analysis for top keywords (in terms of frequency) extracted for ten computer science related venues. We can summarize the prominent observations from the experiments as follows:

- For the average of all the venues, paper embedding based models (Model 4, 5, and 6) tend to give the highest performance especially in terms of macro averaged F1-score.
- In general, paper embedding based models (Model 4, 5, and 6) perform better than the word embedding based models (Model 1, 2, and 3).
- When Model 1 and Model 2 are compared, it is seen that the generating summary representation of the paper collections for the observation period improves the trend prediction performance considerably.
- When we compare similarly structured models, Model 2 and Model 3, it is observed that applying CNN did not bring an advantage to generate year based summary of the paper collections.
- Among the baseline methods, logistic regression gives higher prediction accuracy especially in venues AAAI, CIKM and VLDB. Although simple baseline models tend to perform good in some venues, the average results of all venues show that they fail to generate any prediction for some of the labels, as can be seen as NaN in the results such as in Table 4.4.
- When trend prediction is performed by combining the paper collections of all the venues used in the experiments, the gap between the prediction performance of the proposed neural models and the baselines becomes more clear in favor of the proposed models. Under higher volume of paper collections and more amount of evidence, Model 2 provides the best scores. This gives a hint that word embedding based representations become more effective compared to paper embeddings as the size of the data collection increases. On the other hand, for prediction of *Increase* label, Model 6 gives the best performance in terms of precision and F1-score.

As the future work, one possible research direction is to work with text processing and segmentation models. These can help performing queries beyond single keyword. Neural model architectures can be further extended with attention techniques and experimented upon.

CHAPTER 5

CONCLUSION

Scientific research and academic collaboration venues fosters interaction between diverse fields and technique exchanges. Topics of the works presented evolve accordingly and prediction techniques continue to be valuable tools for participants in all levels.

In this dissertation, two complementary problems are chosen to study topic prediction both at individual level and community level. At the individual level we explored the area of information flow and modeled our problem on the receiving end for an author to predict next topic adoption to explore. We complement that study with exploring topic trend prediction for select venues or multiple venues as a whole.

First, we have examined the received influence with the concept *influencee score* and constructed our hypothesis on received influence for a topic will effect the topic adoption decision. Inspired by the work in [29], proposed *influencee score* calculation addresses the efficiency by handling the social stream in backwards. This feature is tested on the framework presented in [11] and Arnet Miner data is used to drive these experiments. Our proposed feature boosts the outcome when combined with the features present in [29]. Both the prediction and recall value benefited from the new feature where recall values enjoyed the effect most.

Second, predicting the trends in academic topics problem is studied. Our main motivation was to utilize deep learning architectures on scientific publication data without the need of hand-crafted features. For this purpose, Microsoft Academic Graph data is used and experiments are conducted on ten computer science venue data that is basically paper title and abstracts. With this sequence data at hand, deep learning

models utilized LSTM and convolution modules in various arrangements. Proposed solutions can be grouped into 2 groups such as handling the data in keyword level via word embedding vectors or paper level via document embedding vectors. In order to baseline every experiment, only GloVe is used as the word embedding solution without any further tuning during model learning. Document level proposed model is also challenged against the Specter [31] and Doc2vec [30] implementations.

We have various observations on our experimental results for topic trend prediction problem. In general and when averaged on the venue performances, paper embedding based models (Model 4, 5, and 6) performed better than the word embedding based models (Model 1, 2, and 3). When Model 1 and 2 compared, introduction of yearly summaries improved the model performance. When comparing the use of Convolutional Neural Network at the initial data handling levels between Model 2 and Model 3, no discriminative performance benefit observed. Although baseline models can show higher performances in some venues, they can not express some labels at all for other venues and averaged results fall behind. When the trend prediction tested against with bigger dataset as all venues combined, our proposed neural models achieve better results than baseline models when compared to single venue experiments. Model 2 achieved best recall and F1-score results for venue averages and in the last all venues combined experiment as can be seen in Table 4.3 and Table 4.4.

Some open problems that worth further investigation can be listed as follows:

- Introducing topic adoption features that incorporates venue information in scientific collaboration networks.
- Topic popularity prediction can be extended with text processing and segmentation models.
- Efficiency of different word vector representations can be explored.
- Attention based deep learning models can be explored to introduce new model architectures
- Graph based algorithms can be studied and benchmarked against deep learning models.

- Effect of citation information inclusion or other academic social network properties can be explored.

REFERENCES

[1] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu, "Clvsa: A convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3705–3711, AAAI Press, 2019.

[2] Q. Li, J. Tan, J. Wang, and H. Chen, "A multimodal event-driven lstm model for stock prediction using online news," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[3] T. Yang, L. Sha, J. Li, and P. Hong, "A deep learning approach for covid-19 trend prediction," *arXiv preprint arXiv:2008.05644*, 2020.

[4] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[5] J. Srivastava, "Data mining for social network analysis," in *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*, pp. xxxiii–xxxiv, IEEE, 2008.

[6] P. Domingos, "Mining social networks for viral marketing," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 80–82, 2005.

[7] C. C. Chen, S.-Y. Shih, and M. Lee, "Who should you follow? combining learning to rank with social influence for informative friend recommendation," *Decision Support Systems*, vol. 90, pp. 33 – 45, 2016.

[8] H. Bagci and P. Karagoz, "Context-aware friend recommendation for location based social networks using random walk," in *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW'16 Companion, pp. 531–536, 2016.

[9] S. A. Rios, F. Aguilera, J. D. Nunez-Gonzalez, and M. Grana, "Semantically enhanced network analysis for influencer identification in online social networks," *Neurocomputing*, vol. 326-327, pp. 71 – 81, 2019.

[10] K. Li and S. Wang, "A network accident causation model for monitoring railway safety," *Safety Science*, vol. 109, pp. 398 – 402, 2018.

[11] D. Yang, Y. Xiao, B. Xu, H. Tong, W. Wang, and S. Huang, "Which topic will you follow?," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7524 LNAI, no. PART 2, pp. 597–612, 2012.

[12] M. E. J. Newman, "Coauthorship networks and patterns of scientific collaboration," *Proceedings of the National Academy of Sciences*, vol. 101, no. Supplement 1, pp. 5200–5205, 2004.

[13] Q. He, B. Chen, and C. L. Giles, "Detecting Topic Evolution in Scientific Literature : How Can Citations Help ?," *Cikm*, pp. 957–966, 2009.

[14] C. X. Lin, Q. Mei, B. Zhao, and J. Han, "PET : A Statistical Model for Popular Events Tracking in Social Communities," *KDD10*, pp. 929–938, 2010.

[15] Z. Xie, "A prediction method of publication productivity for researchers," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 2, pp. 423–433, 2021.

[16] M. Zhou, H. Dong, B. Ning, and F.-Y. Wang, "Recent development in pedestrian and evacuation dynamics: Bibliographic analyses, collaboration patterns, and future directions," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1034–1048, 2018.

[17] S. Yu, F. Xia, and H. Liu, "Academic team formulation based on liebig's barrel: Discovery of anticask effect," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1083–1094, 2019.

[18] V. Prabhakaran, W. L. Hamilton, D. McFarland, and D. Jurafsky, "Predicting the rise and fall of scientific topics from trends in their rhetorical framing," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1170–1180, 2016.

[19] C. Chen, Z. Wang, W. Li, and X. Sun, "Modeling scientific influence for research trending topic prediction.," in *AAAI*, pp. 2111–2118, 2018.

[20] E. Tattershall, G. Nenadic, and R. D. Stevens, "Detecting bursty terms in computer science research," *Scientometrics*, vol. 122, no. 1, pp. 681–699, 2020.

[21] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, 12 2019.

[22] J. L. Hurtado, A. Agarwal, and X. Zhu, "Topic discovery and future trend forecasting for texts," *Journal of Big Data*, vol. 3, no. 1, 2016.

[23] M. Krenn and A. Zeilinger, "Predicting research trends with semantic and neural networks with an application in quantum physics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 117, no. 4, pp. 1910–1916, 2020.

[24] M. Xu, J. Du, Z. Guan, Z. Xue, F. Kou, L. Shi, X. Xu, and A. Li, "A multi-rnn research topic prediction model based on spatial attention and semantic consistency-based scientific influence modeling," *Computational Intelligence and Neuroscience*, vol. Article ID 1766743, no. n/a, pp. 1–15, 2021.

[25] Z. Liang, J. Mao, K. Lu, Z. Ba, and G. Li, "Combining deep neural network and bibliometric indicator for emerging research topic prediction," *Information Processing & Management*, vol. 58, no. 5, p. 102611, 2021.

[26] M. Xu, J. Du, Z. Xue, Z. Guan, F. Kou, and L. Shi, "A scientific research topic trend prediction model based on multi-lstm and graph convolutional network," *International Journal of Intelligent Systems*, vol. n/a, no. n/a, pp. 1–23, 2022.

[27] B. Wang, B. Yang, S. Shan, and H. Chen, "Detecting hot topics from academic big data," *IEEE Access*, vol. 7, pp. 185916–185927, 2019.

[28] A. Dridi, M. Gaber, R. M. A. Azad, and J. Bhogal, "Leap2trend: A temporal word embedding approach for instant detection of emerging scientific trends," *IEEE Access*, vol. 7, pp. 176414–176428, 2019.

[29] K. Subbian, C. C. Aggarwal, and J. Srivastava, "Querying and Tracking Influencers in Social Streams," *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining - WSDM '16*, pp. 493–502, 2016.

[30] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, vol. 32, pp. 1–5, 2014.

[31] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, "SPECTER: Document-level Representation Learning using Citation-informed Transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2270–2282, Association for Computational Linguistics, 2020.

[32] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and Correlation in Social Networks," *SIGKDD*, pp. 7–15, 2008.

[33] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Homophily in Social Networks," *Annu. Rev. Sociol.*, vol. 27, pp. 415–444, 2001.

[34] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback Effects between Similarity and Social Influence in Online Communities," *SIGKDD*, pp. 1–14, 2008.

[35] T. La Fond and J. Neville, "Randomization tests for distinguishing social influence and homophily effects," *Proceedings of the 19th international conference on World wide web - WWW '10*, p. 601, 2010.

[36] S. Aral, L. Muchnik, and A. Sundararajan, "Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21544–21549, 2009.

[37] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration Over Time: Characterizing and Modeling Network Evolution," *Wdsm*, pp. 107–116, 2008.

[38] E. M. Rogers, *Diffusion of Innovations*. New York, NY: Free Press, 5th ed., 2003.

[39] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion

through blogspace," *Proceedings of the 13th conference on World Wide Web - WWW '04*, p. 491, 2004.

[40] M. D. Choudhury, H. Sundaram, A. John, and D. D. Seligmann, "Contextual Prediction of Communication Flow in Social Networks," *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007*, pp. 57–65, 2007.

[41] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst, "Information Propagation and Network Evolution on the Web," *Ml.Cmu.Edu*, no. January 2009, pp. 1–25, 2014.

[42] H. K. Peng, J. Zhu, D. Piao, R. Yan, and Y. Zhang, "Retweet modeling using conditional random fields," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 336–343, 2011.

[43] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," *Proceedings of the 12th {International Conference on Knowledge Discovery and Data mining}*, pp. 44–54, 2006.

[44] X. Cai, Y. Zheng, L. Yang, T. Dai, and L. Guo, "Bibliographic Network Representation Based Personalized Citation Recommendation," *IEEE Access*, vol. 7, pp. 457–467, 2019.

[45] M. Berger, K. McDonough, and L. M. Seversky, "Cite2vec: Citation-Driven Document Exploration via Word Embeddings," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 691–700, 2017.

[46] S. Molaei, H. Zare, and H. Veisi, "Deep learning approach on information diffusion in heterogeneous networks," *Knowledge-Based Systems*, vol. 189, 2020.

[47] M. Yukselen, A. Mutlu, and P. Karagoz, "Infuencee oriented topic prediction: Investigating the effect of influence on the author," *ACM International Conference Proceeding Series*, 2019.

[48] A. A. Salatino, F. Osborne, and E. Motta, "Augur: Forecasting the emergence of new research topics," in *JCDL '18: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pp. 303–312, 2018.

[49] C. Chen, Z. Wang, W. Li, and X. Sun, "Modeling scientific influence for research trending topic prediction," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2111–2118, 2018.

[50] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *KDD '14*, 2014.

[51] R. Caruana, "Learning many related tasks at the same time with backpropagation," in *Advances in neural information processing systems*, pp. 657–664, 1995.

[52] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *CoRR*, vol. abs/1301.3, pp. 1–12, 2013.

[53] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[54] Z. Yao, Y. Sun, W. Ding, N. Rao, and H. Xiong, "Dynamic word embeddings for evolving semantic discovery," *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, vol. 2018-Febua, pp. 673–681, 2018.

[55] M. Chen, "Efficient vector representation for documents through corruption," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–13, 2017.

[56] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[57] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner : Extraction and Mining of Academic Social Networks," *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 990–998, 2008.

[58] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-j. P. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *Proceedings of the 24th international conference on world wide web*, pp. 243–246, ACM, 2015.

[59] M. H. Zweig and G. Campbell, "STATISTICA - ROC curve," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 1993.

[60] L. Roeder, "Netron: Visualizer for neural network, deep learning and machine learning models.." https://www.lutzroeder.com/ai.

[61] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *WWW '15 Companion*, 2015.

[62] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

[63] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[65] I. Pillai, G. Fumera, and F. Roli, "Designing multi-label classifiers that maximize f measures: State of the art," *Pattern Recognition*, vol. 61, pp. 394–404, 2017.

[66] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the fbietkan statistic," *Communications in Statistics-Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.

[67] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, p. 263, International Biometric Soc 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Yükselen, Murat
Nationality: Turkish (TC)

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU Computer Engineering	2008
B.S.	METU Computer Engineering	2005

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2021 - present	Amazon Development Centre Canada	Software Development Engineer
2015 - 2021	Somera A.Ş.	Senior Software Engineer
2014 - 2015	ATOS	Senior System Engineer
2010 - 2014	Environmental Tectonics Corporation	Software Engineer
2008 - 2009	Turkish Naval Forces	Lieutenant
2006 - 2008	TAF-ModSim	Research Assistant
2005 - 2008	METU Computer Engineering	Teaching Assistant
2004 - 2006	Nic.tr TLD	Software Engineer
2003 - 2004	METU Software R&D Center	Software Developer
2002 July	Milsoft A.Ş.	Intern Engineering Student
2001 July	METU-TSK-MODSIMMER	Intern Engineering Student

PUBLICATIONS

International Conference Publications

- Murat Yukselen, Alev Mutlu, and Pinar Karagoz. 2019. "Infuencee Oriented Topic Prediction: Investigating the Effect of Influence on the Author". In Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics (WIMS2019). ACM, New York, NY, USA, Article 14, 9 pages. DOI: https://doi.org/10.1145/3326467.3326488

International Journal Publications

- Murat Yukselen, Alev Mutlu and Pinar Karagoz, "Predicting the Trending Research Topics by Deep Neural Network Based Content Analysis," in IEEE Access, vol. 10, pp. 90887-90902, 2022, doi: 10.1109/ACCESS.2022.3202654

AWARDS AND SCHOLARSHIP

- METU High Performance Award for finishing courses in 1 year with highest GPA (2011)
- TUBITAK 2211 program Ph.D. Scholarship (2013-2014)