This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3202654

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Predicting the Trending Research Topics by Deep Neural Network based Content Analysis

MURAT YUKSELEN[1], ALEV MUTLU[2], AND PINAR KARAGOZ[3],(Member, IEEE)
[1]Department of Computer Engineering, Middle East Technical University, Ankara, Turkey (e-mail: murat.yukselen@ceng.metu.edu.tr)
[2]Department of Computer Engineering, Kocaeli, University, Kocaeli, Turkey (e-mail: alev.mutlu@kocaeli.edu.tr)
[3]Department of Computer Engineering, Middle East Technical University, Ankara, Turkey (e-mail: karagoz@ceng.metu.edu.tr)

Corresponding author: Murat Yukselen (e-mail: murat.yukselen@ceng.metu.edu.tr).

This work is supported by TUBITAK with grant number 117E566.

ABSTRACT Tracking the trends and taking early steps accordingly is important in academia, as well as in other domains such as technology and finance. In this work, we focus on the problem of 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.

INDEX TERMS Trend prediction, keyword popularity prediction, deep learning, document vector, classification, paper embedding.

I. INTRODUCTION

PREDICTING the 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 studies have been conducted to predict trends on a variety of domains including 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 [5]–[8]. As another important problem, predicting the trend of the topics has several benefits for the academic research community. Such insights may help funding agencies to optimize their policies [9] and guide technology companies to shape their policies [10]. In addition, knowing future research trends can be useful for new researchers to plan their studies [11].

With the recent advancements in machine learning, deep learning based solutions have been devised for prediction problems in a variety of domains, 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 [10], [12]. There are recent efforts focusing on research topic prediction [13]–[15] and topic trend prediction [16], 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 [17] rather than trend prediction. In this paper, we aim to explore whether deep neural architectures can be more effective in coping with the 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 [17], [18], 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 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.

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, but this time, explore the use of different paper embedding modules, LSTM based paper embedder module (Model 4), doc2vec (Model 5) and Specter (Model 6).

The performance of the proposed methods are analyzed 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, the prediction performance is additionally explored under a higher volume of publications with more evidence for the trend.

The contribution of this work can be listed as follows:

- The research trend prediction analysis is formally defined through keyword trend prediction over a collection of academic papers.
- A family of neural models is proposed for the defined problem. Each of the proposed neural models explores 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 regression and Support Vector Machine (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 paper is organized as follows. In Section II, an overview of the related studies is presented. In Section III, preliminary information for the proposed method is given. In Section IV, the problem definition and the proposed methods are described. In Section V, experiment setting and comparative prediction performance analysis of the methods are presented. The paper is concluded in Section VI with an overview of the work and the results, as well as the future work.

II. RELATED WORK

In this section, we summarize the related studies under two groups. Firstly we summarize the studies on academic trend prediction. Then, document embedding related studies are briefly described.

A. STUDIES ON ACADEMIC TREND PREDICTION

Prediction tasks within the domain of academic research efforts have been an attractive problem. Citation recommendation [19], scientific document representations [20], [21], cascade prediction [22], and topic diffusion [23] are some of such problems that are studied 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 [12] challenges the problem of topic discovery and trend forecasting from texts. As in our approach, the study uses token simplification methodologies, but at 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 [24], the authors focus on detecting emerging academic topics at an early stage which they call *embryonic phase*. Their method is based on constructing evolutionary topic co-occurence networks on yearly basis and devising a clustering algorithm named *Advanced Clique Percolation Method (ACPM)* for detecting clusters of the topics in the evolutionary networks. After applying filtering on generated clusters, clusters of the collaborating topics that enlarge in increasing pace are denoted as emerging topics. The method is evaluated on a data set where the debuting topics are manually annotated.

In [25], pairwise influences between venues are studied and trending topics, which consist of topical words, are predicted for the next year for the academic publication venues. Each venue is handled as a bag of words and topic embeddings are learned over these words. Recurrent Neural Networks (RNN) is utilised over venue vectors by further considering the influence between them via topic embeddings. The authors quantify venue to venue influence over years and detect trending topical words. Trending topics are determined by comparing the word percentage increase with respect to previous years.

In [17], the authors assess paper content by extracting keywords through *deepwalk* [26] and determine popular topics in the field by examining co-occurrences of detected keywords. For keyword processing, they use pre-trained word2vec embeddings for English papers. In the study, keyword extraction is performed with a feature based approach.

The study in [18] 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 [13], 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 [14] 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 [15] 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 [16], 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 graphs of publications are 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 work, there are studies that use neural network based solutions such as those given in [25], [13], [15] and [16]. However, there are basic differences in problem definition and data modeling. In [25], the focus is on venue to venue influences and the use of venue and topic embeddings. The study in [13] also differs from our work since a scientific semantic network is employed and the analysis is conducted on this network. Although the problem definition given in [13] has similarity to the problem definition of our work, a wider prediction time window of 5 years is considered in [13] . In [15], topics are represented with feature vectors where several of the features, such as Web of Science categories, are possibly extracted from external resources. In [14] and [16], the problem modeling differs 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 [9] and [15] where they start with a predefined set of topics. Then the approach in [9] uses clustering for 500 topics and the study in [15] uses statistical techniques to generate topics with foreground and background corpus. However model outputs are not compatible as the study in [9] labels only bag of word

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3202654

topics to increase and decrease whereas the one in [15] ranks the candidate bag of word topics. Experiments presented in [9] similarly utilize the last 5 years as an observation window. The problem definition and model outputs are similar in [10] and [16], 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.

B. 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 a 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 [27].

Word analogy task described in [28], known as *word2vec*, enables similarity and analogy calculations by vector operations. As a pre-trained embedding collection, GloVe [29] is popularly used as it aims to solve the shortcomings of locality of skip-gram based training. Main approach in GloVe is to incorporate the 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 [30] 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 with the main goal to query with the help of word embedding. In [31], word2vec is further extended to doc2vec in order to generate *paragraph vectors*. In [32], 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 [21] 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 [20] is a deep learning transformer model specialized for scientific papers, which uses citation information in the training phase and can produce stable document embedding afterwards.

In our work, 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 keywords.

III. PRELIMINARIES AND PROBLEM DEFINITION

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

Definition 3.1 (Query Keyword): Query keyword q is a term or a token that has significance for a research topic.

Definition 3.2 (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 3.3 (Observation window): Observation Window is a time range consisting of a 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.

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 3.4 (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.

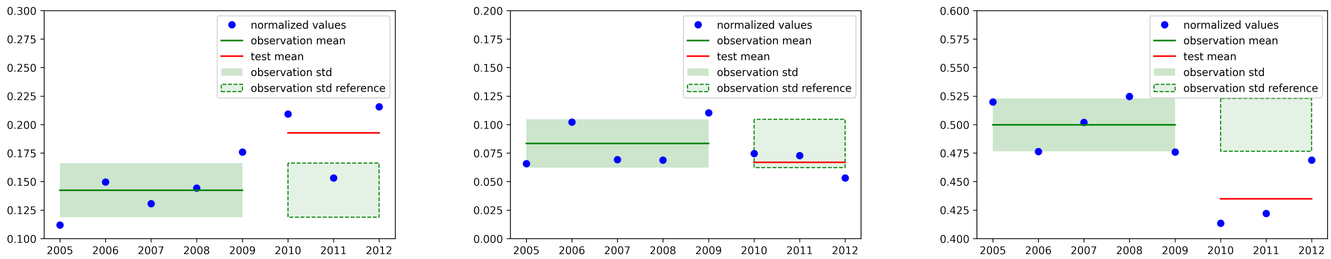
In our work, the prediction window length is set as 3 on the basis of validation analysis. This expresses that the trend label prediction is made for the subsequent 3 years.

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 3.5 (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 3.6 (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 1 illustrates the trend labels on a sample case. In Figure 1a, the trend is labeled as *Increase* since avg_{pred} lies above the observation distribution. Similarly, Figure 1b illus-



(a) Increase (b) Steady (c) Decrease

FIGURE 1. Illustration of the Keyword Trend Labels. First 5 years is Observation window and last 3 years is Prediction window.

trates the *Steady* label and Figure 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 3.7 (Trend Prediction): Given a query keyword, *trend prediction* denotes predicting the label among *Increase*, *Steady* and *Decrease* for the prediction window.

Definition 3.8 (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.

IV. 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 paper collection 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 ways 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 solve 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 vectors and outputs the resulting vector.
Dense: Dense block connects each input to each output, forming a 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 a 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.

A. 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 1, with *n* title tokens and *k* abstract tokens.

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

Depending on the model, either the embedding of the tokens or the whole document is constructed and used. The papers are stacked yearly as in Equation 2 and zeros are padded to $maxlen$ of the longest $TokenStream$.

$$TokenStream_{year} = [paper_{year,1}, ..., paper_{year,l}] \tag{2}$$

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 3. However it can be adjusted for any size of observation window.

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

For a query token on a given $year$, data samples used for training can be defined as in Equation 4. Query is a single token keyword whose trend is to be predicted. $label$ belongs to set {-1, 0 , 1} for a regression task or set {0, 1, 2} for a classification task.

$$x, y = [Year - Stream_{year}, query], label \quad (4)$$

B. WORD EMBEDDING BASED NEURAL MODELS

The 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 a one-year stream to construct a year summary vector (shown in Figure 2[1]). As shown in the figure, the size of the input 2-D array to the models is *5 years x 60919 tokens per year*[2]. The embedding layer generates embeddings of 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 a collection of papers per year into a yearly summary vector of size 32 (shown in Figure 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

[1]Model figures are generated via Netron application [33].

[2]Note that the ? mark in the main input size in Figure 2 denotes the number of input batches

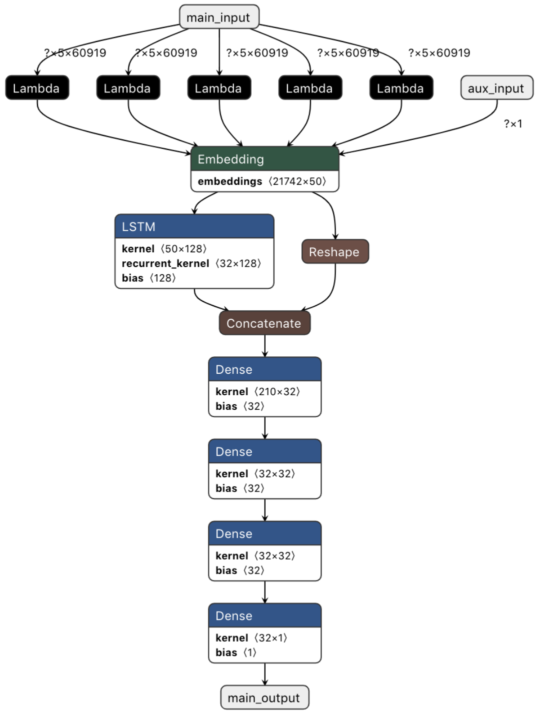


FIGURE 2. Model 1: The architecture includes a shared LSTM module to process paper collections per year, yearly embeddings are concatenated

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 a shared *convolution* layer to generate yearly paper embeddings (given in Figure 4). The convolution layer applies double 1D-convolution with kernel size 5 and 64 filters. After applying a 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 the 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.

C. PAPER EMBEDDING BASED MODELS

The paper embedding based models consider a paper as a whole and process each paper individually. Figure 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 paper embedding construction, we use three alternative methods: pre-trained LSTM (*Paper Embedder LSTM*), doc2vec [31], and Specter [20]. After generating a

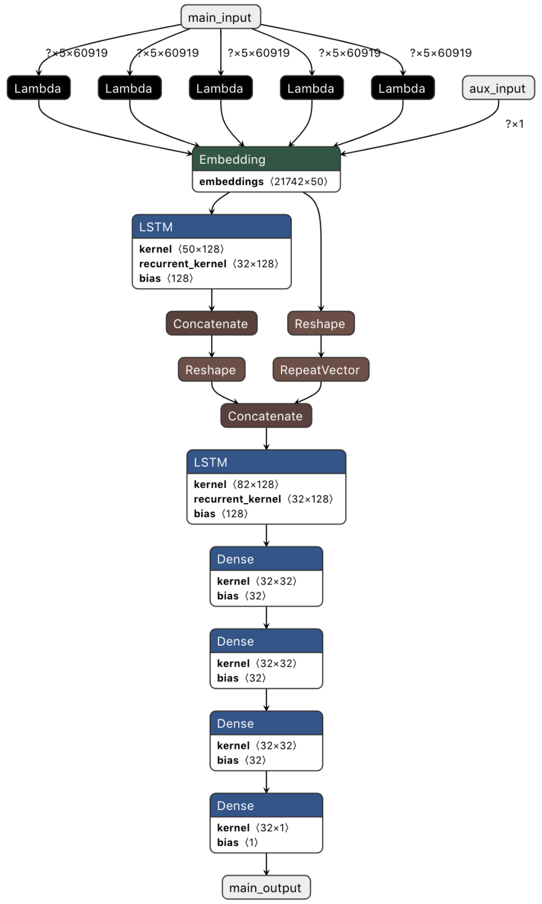


FIGURE 3. Model 2: The architecture includes a shared LSTM module for each year, it includes another LSTM over all year results

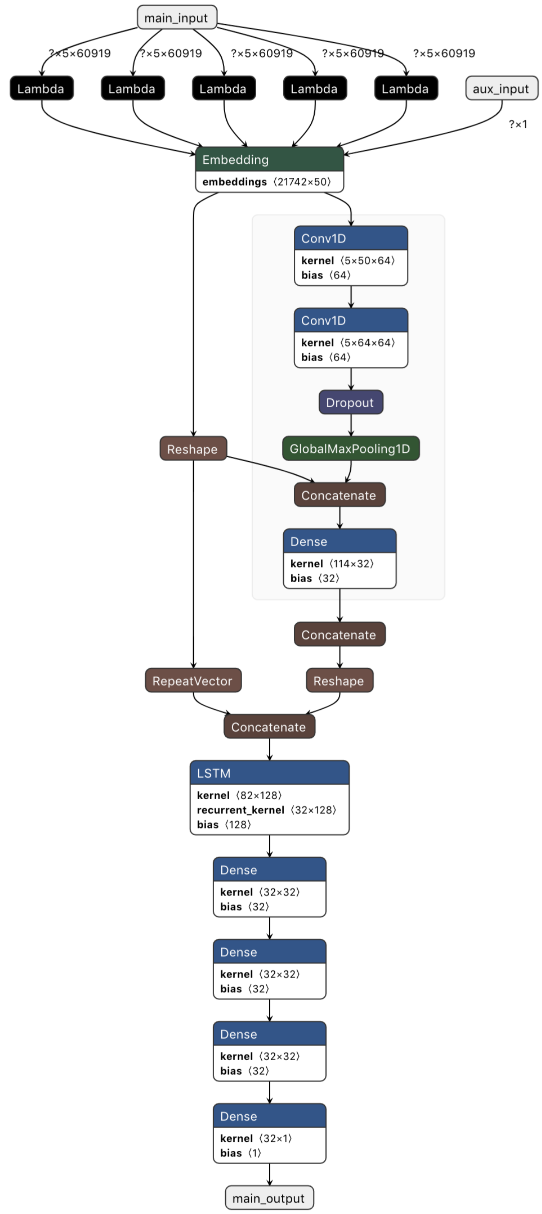


FIGURE 4. Model 3: The architecture includes a Convolution module for each year, it has an LSTM module to process all year results

paper vector, a shared LSTM module is utilized to generate the yearly summary vector. As the last 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 5). Its result is fed into a 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* [31] generates paragraph vectors through training on word vectors. On the other hand, *Specter* [20] 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 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 in a 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

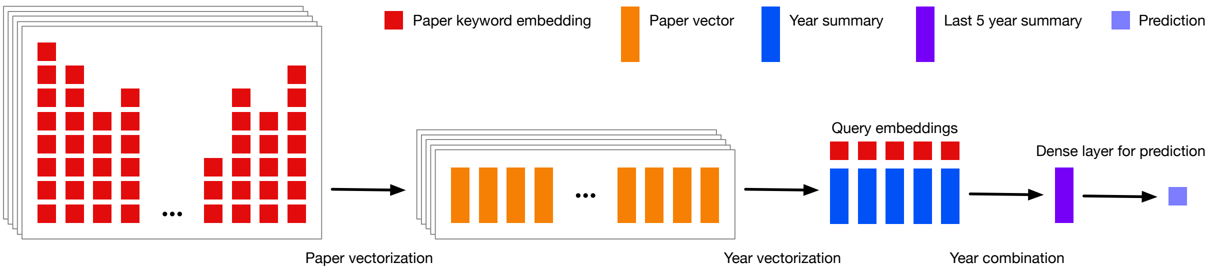


FIGURE 5. Data flow for processing papers to a vector

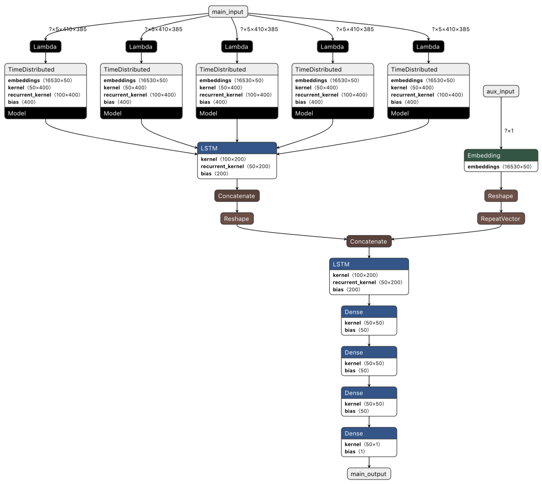


FIGURE 6. Model 4: The model includes *Paper LSTM module* to generate paper embeddings, LSTM module to generate one-year summary embeddings, and another LSTM module to generate representation over the observation window.

concatenation of yearly summary vectors and the embedding of the query keyword. 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 a dense layer to generate the trend prediction of the query keyword.

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 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

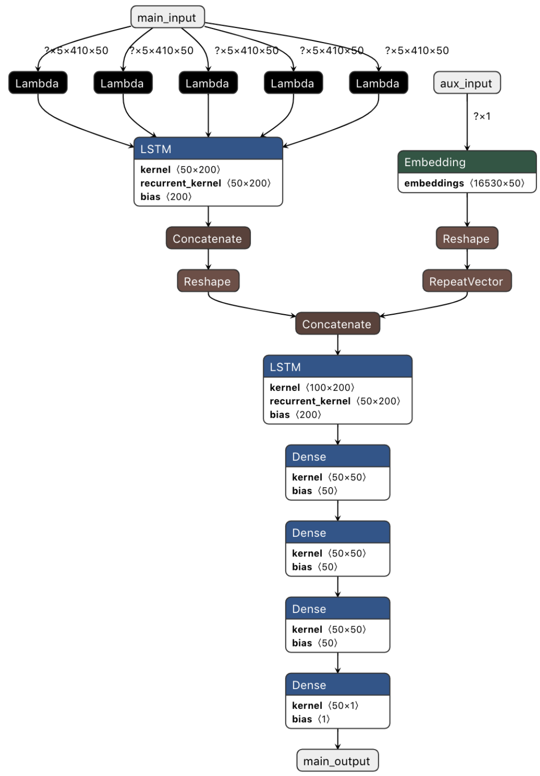


FIGURE 7. Model 5 & 6: The model includes Doc2Vec (in Model 5) or Specter (in Model 6) for each paper, an LSTM module to generate one-year summary representation, and another LSTM module to generate representation over the observation window.

Specter for constructing the paper vectors. Figure 7 shows the model.

V. 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 (venues) 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.

A. DATA SET AND EXPERIMENT SETUP

As the data set, we use a collection of papers obtained from Microsoft Academic Graph [34]. 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.

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 in alphabetical order in Table 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[3]. Stopwords are also removed from the keyword set.

To be used as query keywords in the training, validation and testing phases, we start with top 500 keywords which are chosen according to the frequency. Then for each year, we arbitrarily keep a balanced keyword collection in terms of our labels decline, steady and increase. 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 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 3.6. 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 3.6, 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.

B. IMPLEMENTATION SETUP

For the word embedding based models, we use pre-trained word embedding weights from a ready source, GLoVe [29].

[3]https://spacy.io/

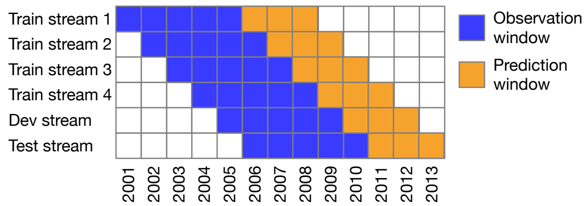


FIGURE 8. Illustration of data set splits and time windows

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 [35] is used. For Specter paper embedding model, implementation provided on Github [4] is used. The proposed deep learning architectures, LSTM and CNN modules are developed by using the Keras library [36].

C. BASELINE MODELS

We adopted four baseline methods, two regression based and two classification based ones. Linear regression and Support Vector Regression (SVR) are used as the regression based baseline methods. For classification based baseline methods, Logistic regression (with class labels) and Support Vector Classification (SVC) 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 II, 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 [37] package.

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 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.

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

D. 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 the macro averaged setting as it gives equal weights to each class [38].

[4]https://github.com/allenai/specter

TABLE 1. Selected venues in the data set

Short form	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

TABLE 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 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

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. 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 dataset. 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 present 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 2, the precision results are given. The results indicate that the baseline models achieve the highest scores for 6 venues out of 10. However, a vast number of NaN 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 models (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 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

TABLE 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

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.

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, 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.

E. 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 employed Iman-Davenport test [39] and Nemenyi post-hoc test [40]. 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. The 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 the 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 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.

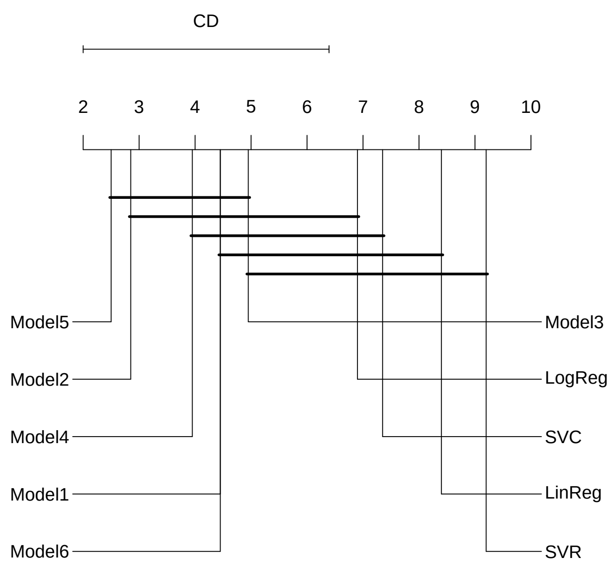


FIGURE 9. Nemenyi post-hoc test result for Model $F1$ macro averaged score

F. LABEL-WISE KEYWORD TREND PREDICTION PERFORMANCE ANALYSIS

Since predicting the topics having an increasing trend has more importance in practical use, we further report and analyze the precision, recall, and F1-score values for the label *Increase*. Table 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.

TABLE 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 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

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 6, we report the recall results for label *Increase*. The results are similar 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 7, the macro averaged F1-scores for the *Increase* label are reported. As seen in the table, the neural models rank 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*.

G. STATISTICAL SIGNIFICANCE ANALYSIS OF LABEL-WISE PERFORMANCE RESULTS

We statistically analyze the F1-scores for the label *Increase*, as well. The Iman-Davenport test gives p-value = 4.448e-11, which indicates that methods statistically differ. Figure10 depicts the Nemenyi post-hoc test results. As the result indicates, Model 2 has the highest mean average rank and statistically differs 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.

H. 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 dataset 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 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.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3202654

TABLE 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

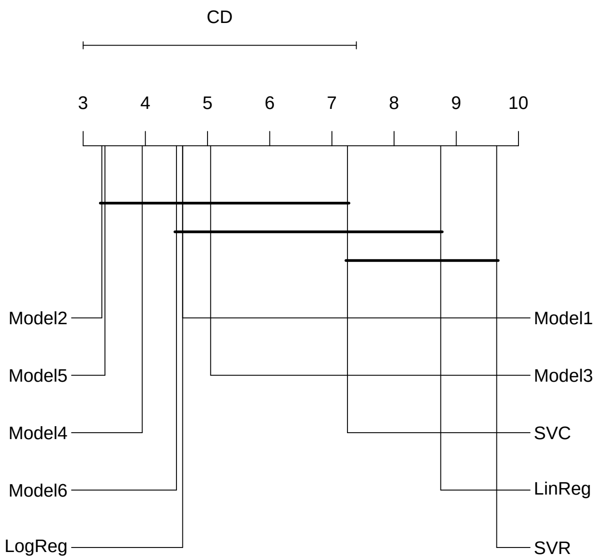


FIGURE 10. Nemenyi post-hoc test result for Label *Increase* $F1$ score

TABLE 8. Field of Study Classification results on paper embeddings

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

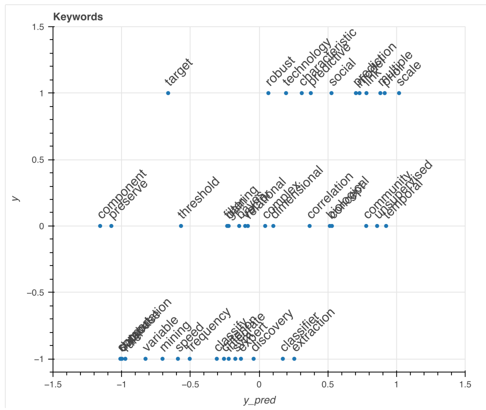


FIGURE 11. Keyword plot for the ICDM

TABLE 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	

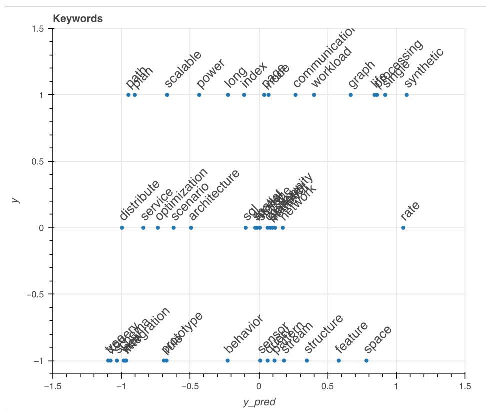


FIGURE 12. Keyword plot for the VLDB

I. 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 11, Figure 12 and Figure 13 for ICDM, VLDB and WWW, respectively.

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

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3202654

TABLE 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	

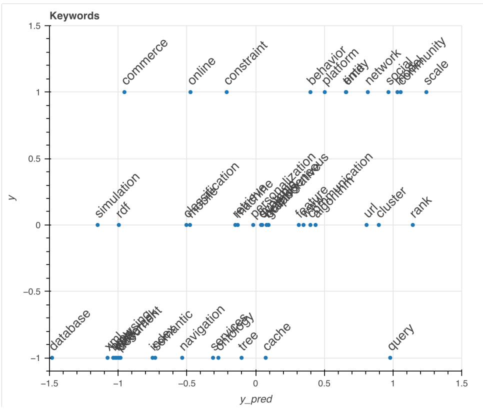


FIGURE 13. Keyword plot for the WWW

TABLE 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	

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 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 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 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 modeling 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 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 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 11.

VI. CONCLUSION

In this work, 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 handcrafted features and additional information other than paper collections. Instead of a 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, the 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.
- 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 result suggests 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 the 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.

ACKNOWLEDGMENTS

This work is supported by TUBITAK with grant number 117E566.

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*. AAAI Press, 2019, pp. 3705–3711.

[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] F. Gurcan and N. E. Cagiltay, "Exploratory analysis of topic interests and their evolution in bioinformatics research using semantic text mining and probabilistic topic modeling," *IEEE Access*, vol. 10, pp. 31 480–31 493, 2022.

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

[7] 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.

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

[9] 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)*, 2016, pp. 1170–1180.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] B. Wang, B. Yang, S. Shan, and H. Chen, "Detecting hot topics from academic big data," *IEEE Access*, vol. 7, pp. 185 916–185 927, 2019.

[18] 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. 176 414–176 428, 2019.

[19] 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.

[20] 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*. Association for Computational Linguistics, 2020, pp. 2270–2282. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.207

[21] 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.

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

[23] 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.

[24] A. A. Salatino, F. Osborne, and E. Motta, "Augur: Forecasting the emergence of new research topics," in *JCDL âĂŹ18: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, 2018, pp. 303–312.

[25] 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.

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

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

[28] 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. [Online]. Available: http://arxiv.org/abs/1301.3781

[29] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language*

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3202654

Processing (EMNLP), 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[30] 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.

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

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

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

[34] 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.

[35] 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: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

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

[37] 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.

[38] 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.

[39] 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.

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

MURAT YUKSELEN received the B.S. and M.S. degrees from Computer Engineering Department, Middle East Technical University (METU) in 2005 and 2008. He is currently pursuing the Ph.D. degree in the same department. His research interests include Machine Learning, Data Mining and Multi-Agent Systems.

ALEV MUTLU is an assistant professor at Kocaeli University. He received his BS degree from Kocaeli University and Ph.D. degree from Middle East Technical University (METU), all in computer engineering. He has spent a year at Case Western Reserve University, OH, USA as a visiting researcher. His research focuses on multi-relational data mining with particular interest in concept discovery and concept discovery in graph databases.

PINAR KARAGOZ Dr. Pinar Karagoz is Professor in Computer Engineering Department, Middle East Technical University (METU). She received her Ph.D. from the same department in 2003. She worked as a visiting researcher in State University of New York (SUNY) at Stony Brook. Her research interests include data mining, web usage mining, social network analysis, information extraction from the web, semantic web services, web service discovery and composition. Dr. Karagoz has authored publications in international journals and leading conferences. Some of her papers were published in journals such as IEEE TKDE, IEEE Industrial Informatics, ACM TWEB, Information Systems Journal, SIGMOD Record, Knowledge and Information Systems, Knowledge based Systems. Some of her research were presented and published in conferences including VLDB, CIKM, ASONAM, DAWAK, ICWS. She has also taken part in the organization committee of several conferences including ICDM and VLDB.