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

Leap2Trend: A Temporal Word Embedding Approach for Instant Detection of Emerging Scientific Trends

AMNA DRIDI[1], MOHAMED MEDHAT GABER[1], R. MUHAMMAD ATIF AZAD[1], AND JAGDEV BHOGAL.[1]

[1]School of Computing and Digital Technology, Birmingham City University, Millennium Point, Birmingham, B47XG, UK

Corresponding author: Amna Dridi (e-mail: amna.dridi@mail.bcu.ac.uk).

The work of A. Dridi was supported by the Faculty of Computing, Engineering and Built Environment, Birmingham City University, through a Full Bursary Ph.D. Scholarship.

ABSTRACT Early detection of emerging research trends could potentially revolutionise the way research is done. For this reason, trend analysis has become an area of paramount importance in academia and industry. This is due to the significant implications for research funding and public policy. The literature presents several emerging approaches to detecting new research trends. Most of these approaches rely mainly on citation counting. While citations have been widely used as indicators of emerging research topics, they suffer from some limitations. For instance, citations can take months to years to progress and then to reveal trends. Furthermore, they fail to dig into paper content. To overcome this problem, we introduce *Leap2Trend*, a novel approach to instant detection of research trends. *Leap2Trend* relies on temporal word embeddings (*word2vec*) to track the dynamics of similarities between pairs of keywords, their rankings and respective uprankings (ascents) over time. We applied *Leap2Trend* to two scientific corpora on different research areas, namely *computer science* and *bioinformatics* and we evaluated it against two gold standards *Google Trends hits* and *Google Scholar citations*. The obtained results reveal the effectiveness of our approach to detect trends with more than 80% accuracy and 90% precision in some cases. Such significant findings evidence the utility of our *Leap2Trend* approach for tracking and detecting emerging research trends instantly.

INDEX TERMS Citation counts, Google Scholar, Google Trends, Temporal word embedding, Trend analysis

I. INTRODUCTION

With the increased use of scholarly networks and digital libraries, we now have a huge amount of scholarly documents exceeding 114 million accessible on the web on 2014 with a rate of tens of thousands per day for newly generated scholarly documents according to [1]. This has recently brought great interest to mining scholarly data and revealing emerging trends. For instance, we have witnessed the emergence of several studies [2]–[10], discussing how to predict future research trends and popular research topics, termed as *trend analysis* approaches [11].

Most of these approaches rely mainly on citation counting from papers which have been published, and consequently find clues to topic evolvement [12]. While citation counts are used as indicators of emerging research topics, they can take months or even years to reveal research trends. Also, they fail to dig into the paper content. Therefore, there is a need to shift from citation-based approaches to more instant yet accurate approaches for trend analysis that drill into the content of scholarly publications.

Following this trend, some work [7], [13]–[16] emerged and explored text mining techniques, namely topic models, to forecast the emergence of new research topics. While topic models intend to extract semantics by capturing document level associations between words, they fail to detect pairwise associations of keywords. This is a considerable limitation since emerging topics often start first by an increasing closeness of keywords that may lead to a merge. For instance, the research topic *"deep learning"* resulted from the merge between the two keywords/topics *"machine learning"* and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

"neural networks". For this reason, a fine-grained study of the associations between words is needed for an instant detection of emerging trends.

In this paper, we present *Leap2Trend*, a novel approach that aims to effectively and instantly detect the emerging scientific trends. *Leap2Trend* addresses the limitation of topic models by proposing *word embeddings* – which are neural network based techniques in natural language processing – due to their abilities to detect pair-wise associations between words. Indeed, word embeddings can successfully capture both the semantic and the syntactic features of words [17]. More specifically, *temporal word embeddings* have been adopted to learn distributed vector representations of keywords over time. To do so, we repeatedly train each corpus $P_t, t = 1, \ldots, T$ that corresponds to the corpus of all research papers in the t-th timespan. This helps to capture the fast converging keywords which could lead to emerging scientific trends. Hence, temporal word embeddings could represent an effective tool to detect evolving scientific trends/keywords instantly.

From this perspective, *Leap2Trend* operates in four main stages. First, it trains temporal embeddings [18] following two temporal paradigms: incremental and sliding, in order to study the impact of research history in detecting new emerging trends. After each training, *Leap2Trend* creates a *similarity matrix* that stores the similarities of potential pairs of keywords of interest – these keywords of interest represent the top frequent keywords that appear in the titles of publications. Then, it proceeds to rank this matrix and compute the ascents in ranking over different timespans. Finally, for each picked jump, *Leap2Trend* performs different evaluations against *Google Trends hits* and *Google Scholar citations* in order to test if the detected jump of the pair of keywords refers to a new emerging topic. The obtained results on NIPS and MICCAI – two datasets on two different research areas, namely *computer science* and *bioinformatics* – show the effectiveness of our proposed approach to detect new emerging trends with high accuracy and precision exceeding 80% and 90% respectively in some cases.

The main contributions of this paper are fourfold:

1) Introducing *Leap2Trend*, a new framework for the detection of new research trends at a very early stage;
2) Leveraging word embedding techniques, namely word2vec [19] for fine-grained content analysis of scientific corpora;
3) Applying *Leap2Trend* to real-world datasets in two different research areas – computer science and bioinformatics – which could give insights about the validity and the generalisability of our approach;
4) Validating the approach using Google Trends hits and Google Scholar citations as gold standards.

The rest of the paper is organised as follows. Section II outlines existing work on trend analysis and temporal word embeddings. Section III introduces *Leap2Trend* and details its different stages. Section IV describes the used datasets, presents the gold standards, reports and discusses the experimental results. Finally, we conclude and point to future directions in Section V.

II. RELATED WORK

In this section, we first review methods on trend analysis in big scholarly data, and then discuss works on temporal word embedding, which is an important component in our approach.

A. TREND ANALYSIS IN BIG SCHOLARLY DATA

The analysis of big scholarly data (ABSD) has attracted considerable interest in the past few years [11], [20]–[25] due to the explosion of publicly available data on scholarly networks and digital libraries. One of the main tasks of ABSD is to study how research topics evolve over time and to track emerging topics and trends. In the literature, this task is termed as *trend analysis*.

Different approaches in the literature dealt with trend analysis using different features such as citation counts, paper content especially keywords, or both of them. We can then categorise these approaches into three categories with respect to the features they have been using: (i) bibliometrics-based approaches [2], [5], [6], [8], [10] that are based on social network analysis and citation analysis; *content-based approaches* [9] that treat entities – essentially keywords – reflecting the paper content [9]; and *hybrid approaches* [3], [4], [26] that combine both citation and content.

These approaches have been applied to a wide range of disciplines such as relations and economy [8], innovation and entrepreneurial ecosystem [10], business [6] and business model innovation [2], marketing and tourism [9], medical domain [27], biology [28], information science [5] and computer science [3], [4], [29]–[33].

In this paper, we concentrate on the areas of computer science (CS) and bioinformatics. While, we are not aware of previous work on predicting research trends in bioinformatics, we report research findings on trend analysis within CS. For instance, Hoonlor et al. [4] were the early researchers interested in learning about the evolution of CS research. They analysed data from 1990 to 2010 on proposals for grants supported by the U.S National Foundation[1] and on CS publications in the ACM Digital Library[2] and IEEE Xplore Digital Library[3] using sequence mining, bursty word detection and clustering, network extraction and visualisation. They aimed to investigate changes over time in the CS research landscape; interaction of CS research communities; similarities and dissimilarities between research topics. Similarly, Hou et al. revealed the evolution of research topics between 2009 and 2016 using the timeline knowledge map through Document-Citation Analysis (DCA) of 7574 articles published in 10 Information Science (IS) journals including 20960 references. They used dual-map overlays of the IS

[1] https://www.nsf.gov/
[2] https://dl.acm.org/
[3] https://ieeexplore.ieee.org/

literature to track the evolution of the knowledge base of IS research based on scientometric indicators (H-index), citation analysis and scientific collaboration. In the same context, Effendy and Yap [3] performed trend analysis using the Microsoft Academic Graph (MAG)[4] dataset. But, in addition to the bibliometric-basic method (citation analysis), they applied a content-based method by using the hierarchical *FoS (Field of Study)* provided by MAG for each paper to measure the level of interest in any particular research area or topic, and consequently revealed general publication trends, evolution of research areas and the relationship among research areas in CS.

Both approaches described above can be categorised as hybrid approaches. They combine the citation analysis with the content analysis to reveal research trends. The content analysis only studies bursty keywords in [4] and fields of studies in [3] without drilling into the paper content or following a fine-grained analysis. Instead, they focus on citation analysis to reveal citation trends and consequently the evolution of research areas. While citation counts are deemed essential to evaluating the importance of scientific work, the citing behavior could possibly be for non-scientific reasons [34]. Moreover, citations can take months to even years to stabilise enough to reveal research trends. As a matter of fact, there can be interesting papers – termed as *sleeping beauties* [30] – which do not get cited much for several years after publication, but then unexpectedly start getting cited.

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

In this direction, some work has begun to emerge. Anderson et al. [13] have developed a people-centric methodology for computational history that tracks the flow of authors across topics to discern how some sub-fields flow into the next, forming new research directions. This methodology is based on a central phase of topic modelling that classifies papers into topics and identifies the topics the people author in. In the same context, Salatino et al. [7] have proposed *Augur* which is an approach that analyses the diachronic relationships between research areas and detects clusters of topics that exhibit dynamics of already established topics. Similarly, Li et al. [15] have recently proposed an improved method by introducing WordNet to LDA in order to find latent topics of large corpora of NIPS publications and discover the dynamics of research topics. To do so, their method groups the documents by time in each topic. Then, it counts the number of documents by time which helps to reveal whether the topics are rising or falling in popularity.

While these approaches [7], [13], [15] intend to perform a content analysis of research papers by applying topic modelling, they still suffer from the delay time in the detection of trends. For instance, both the flow of authors across topics and the dynamics of established topics take time to happen. In addition, topic modelling – as a text mining technique used for these content-based approaches – is not able to detect pairwise associations between words while the study of these associations could lead to the detection of emerging trends at a very early stage and even instantaneously.

To overcome this problem, we propose in this paper a word embedding technique to dig into the paper content towards an instant detection of emerging trends over time. To the best of our knowledge, the proposed approach represents the first attempt harnessing neural network techniques in trend analysis and it is the first applied to more than one research area, i.e. computer science and bioinformatics. The unique ability of word embeddings to capture fine-grained associations between words makes the comparison of our approach with other approaches such as topic modelling invalid.

B. TEMPORAL WORD EMBEDDINGS

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

Although the study of *temporal word embeddings* is relatively new, some work has emerged [18] on how to leverage word embeddings for time-aware knowledge extraction tasks such as sentiment analysis [35], [36] or temporal information retrieval [37], [38]. In general, the approaches in previous work can be categorised into two main categories according to [18]: *linguistic studies* and *event detection approaches*.

Linguistic studies focus on learning and understanding the semantic shifts of human language in general context. As a matter of fact, these studies aim to (i) explore and analyse emerging word meanings and semantic shifts of particular words [39]–[41] or sentiment words [35], (ii) detect temporal correspondence that requires finding different words with semantically similar meanings at different points in time [42], [43], (iii) identify changes in word usage overtime using word epoch disambiguation [44], [45], and (iv) reveal statistical laws of semantic evolution [46]. While *linguistic studies* attempted to trace temporal changes in language semantics in a general context, *event detection approaches* have been proposed to track the 'cultural' semantic shifts that follow real-world events such as tracing armed conflicts [47], performing a time-sensitive query expansion for temporal information retrieval [38] or detecting trending concepts behind words [48].

Following this trend, our work tends to be placed where both categories will be employed to trace evolving keywords in *scientific language* in order to detect trending scientific topics. In this paper, we concentrate on the area of computer

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

science and bioinformatics. To the best of our knowledge, the proposed approach represents the first attempt harnessing *temporal word embeddings* in a domain-specific language – *scientific language*, aiming to instantly detect emerging keywords in the area of computer science and bioinformatics.

III. LEAP2TREND

In this study, we present *Leap2Trend* which is a novel approach for an effective and instant detection of emerging scientific trends. *Leap2Trend* follows a fine-grained text mining approach that digs into textual content of research papers and grasps semantics by applying neural network based technique, namely word embeddings (word2vec) [19]. Accordingly, we adopt a temporal word vector representation that learns temporal embeddings and tracks the dynamics of keywords over time in order to capture the fast converging keywords which could led to emerging scientific trends.

The workflow of *Leap2Trend* is depicted in Figure 1 and it follows four stages:

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

ii **Word embeddings.** In this stage, we apply word2vec embedding model with its *skip-gram* architecture [49] to learn the distributed vector representations of keywords over time. This stage is repeated for each corpus $P_t, t = 1, \ldots, T$ that corresponds to the corpus of all research papers in the t-th timespan.

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

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

In the next sections, we will detail the functionalities of these stages.

A. DATA PREPROCESSING

1) Language-based preprocessing

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

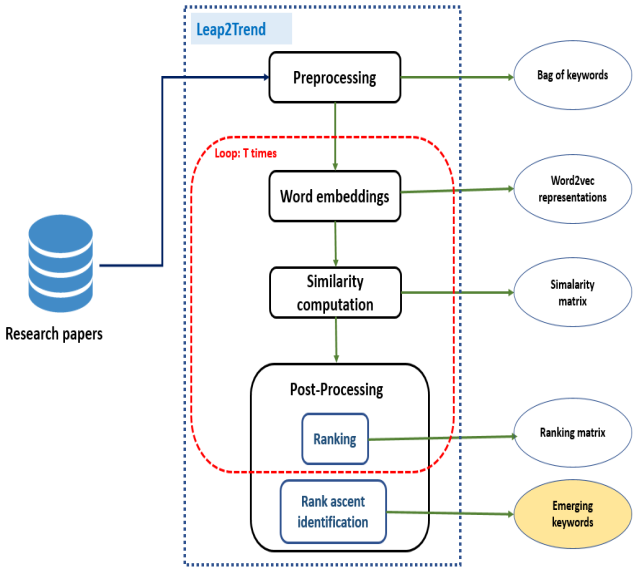


FIGURE 1: Workflow of Leap2Trend

i We remove stop words from the vocabulary using Stanford NLP stop word list[5] enriched by a list of 170 academic stop words that we defined from common academic vocabulary like "*introduction, abstract, conclusion, table, figure, etc.*"

ii We construct a bag of keywords where keywords are either *unigrams* or *bigrams* extracted with *word2phrase*. *Word2phrase* is a word2vec package tool that compoundifies n-grams in a text corpus based on a minimum and a maximum frequency [19]. To compute n-grams, we need to run *word2phrase* $(n-1)$ times successively. In our case, we ran it only one time as we need a vocabulary of 1-grams (unigrams) and 2-grams (bigrams) only. We set the minimum frequency (*min_count* parameter) to 10 in order to remove the infrequent words and reduce the model size. Recall that *trigrams* are also important and highly used in the area of computer science such as "artificial neural netoworks", "support vector machines", *etc*. But, in general their abbreviations are more frequent ("ANN", "SVM"). For this reason, we did not consider trigrams in our analysis. However, we consider their abbreviations as unigrams.

2) Time-based preprocessing

After performing a language-based data preprocessing stage, we proceed with a time-based data processing step that aims to divide the scientific corpora P into T timespans denoted by $P = (P_1, \ldots, P_T)$ where each $P_t, t = 1, \ldots, T$ is the corpus of all research papers in the t-th timespan. This step is important to fulfill the temporality of the task of scientific trend detection and track the evolving keywords over time. To this end, we adopt a dynamic data integration

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

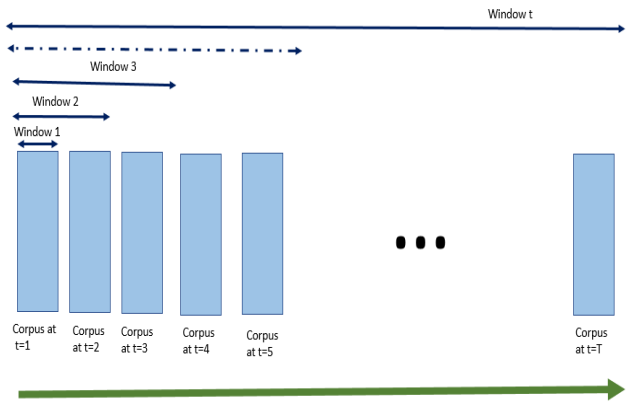


FIGURE 2: Incremental windows

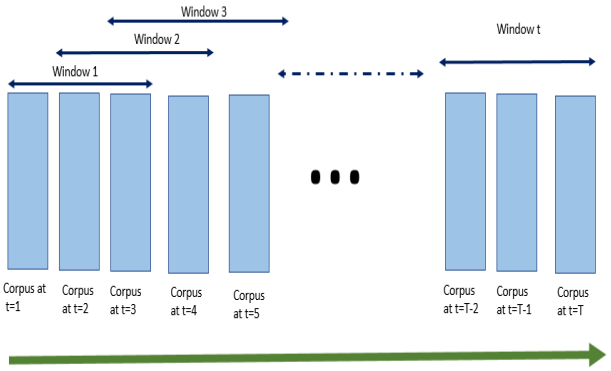


FIGURE 3: Sliding windows

of corpora rather than using static time windows. Our time-based preprocessing has two different temporal paradigms: *incremental windows* and *sliding windows*.

Incremental windows. Each window or timespan t represents a sequence of time stamped corpora $P_t, t = 1, \ldots, T$ gradually created following a 1-year annual basis. Therefore, the corpus of the window $t'_{1 \leq t' \leq t}$ will contain the aggregated corpora of the timespan $(1, t')$ as illustrated in Fig. 2. For instance, if we have scientific corpora dated from 2000 to 2018, the corpus of the window 2008 will contain all corpora between 2000 and 2008. The corpus of the last window T contains all corpora from window 1 to window T.

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

Sliding windows Each window t represents a sequence of three time stamped corpora. The corpus of the window t will contain the corpora of the timespan $(t-1, t+1)$ as shown in Fig. 3. For instance, the corpus of the window 2008 will contain the corpora between 2007 and 2009.

The choice of the timespan length is based on the study performed by Anderson *et al.* [13] on evolving scientific topics. Their investigations showed that the interval of three years was successful to track the flow of scientific corpora.

B. WORD EMBEDDINGS

This study introduces a text mining approach based on word embeddings that tracks emerging scientific keywords at an early stage by capturing the evolution and the movement of keywords over time. Accordingly, we adopt a temporal embeddings technique to learn word vectors in a temporal fashion. Any word embeddings technique could be applied such as *word2vec* [19], *GloVe* [50] and *FastText* [51]. However, we have chosen Word2vec for two main reasons. First, because it is the long standing word embedding technique in the area. Second, because word2vec has performed better in most cases in the comparative study conducted by Wang et *al*. [52]. Relying on this study, we use the skip-gram (SG) neural network architecture of word2vec model as it consistently proved to be experimentally better than the continuous bag of words (CBOW) architecture [17].

1) Skip-Gram Neural Network Model

Skip-gram model has been introduced by Mikolov et *al.* [17] for learning high-quality distributed vector representations. The main idea of *skip-gram* is to predict the *context* given a word w_i. Note that the *context* is a window around w_i of maximum size L that represents the span of words in the text which is taken into account both backwards and forwards when iterating through the words during model training.

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

that describes the connections from the hidden layer to the output layer with each column of $\mathscr{W}'$ representing a word w_i from V_t.

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

- The input word w_i and the output word w_j are one-hot encoded into binary vectors x and y of size V_t.
- The multiplication of the binary vector x and the word embedding matrix $\mathscr{W}$ of size $V_t \times N$ gives the embedding vector of the input word w_i; the i-th row of the matrix $\mathscr{W}$.
- The hidden layer represents the resulting embedding vector of dimension N.
- The multiplication of the hidden layer and the word context matrix $\mathscr{W}'$ of size $N \times V_t$ produces the output one-hot encoded vector y.
- The final output layer applies *softmax function* [19] to compute the probability of predicting the output word w_O given the input word w_I, and therefore:

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

where v_w and ${v'}_w$ are the input and output vector representations of w that correspond to x and y in our case, and W is the number of words in the vocabulary that corresponds to V_t in our case.

- The output context matrix $\mathscr{W}'$ encodes the meanings of words as context.

Hyperparameters. *Skip-gram model* depends on several hyperparameters; some of them crucially impact the quality of embeddings, especially *vector dimensionality* and *context window*. Despite that, the majority of applications that used word embeddings as features computed their vector representations with a default or arbitrary choice of hyperparameters.

Since the optimal hyperparameters are known to be often data and task dependent, we proposed a domain-specific approach to hyperparametrisation [53] for skip-gram. The approach uses the *stability of k-nearest neighbors (k-NN)* of word vectors as the objective to optimise for while learning word2vec hyperparameters. The approach has been detailed in our previous work [53]. The basic idea is the following: embedding quality inevitably depends on tuning hyper-parameters defined previously, namely *vector dimensionality* and *context window*. If we choose accurate values of the tuning hyper-parameters, then we expect that the k similar words to a target word w from different embeddings should be similar. Specifically, we propose to fix one hyper-parameter, tune the second one by trying different values and training the model for each value. After each training, we compute word similarities and we define k-nearest neighboring words. The *k-NN stability* is defined as the overlap rate of similar words resulted from two embeddings with different settings as follows $stability = \frac{\mathcal{S}_{E_h}^{word_i} \cap \mathcal{S}_{E_{h'}}^{word_i}}{k} \times 100$, where $\mathcal{S}_{E_h}$ and $\mathcal{S}_{E_{h'}}$ are two sets of words that are similar to a target word $word_i$ but were produced from two different embeddings E_h and $E_{h'}$ with different hyperparameter values. k is the number of nearest neighbors to $word_i$ given by the cosine similarity. This approach showed that the optimal hyperparameters are respectively 200 and 6 for vector dimensionality N and the context window for scientific corpora. Therefore, the skip-gram model is tuned with these hyperparameters in this work.

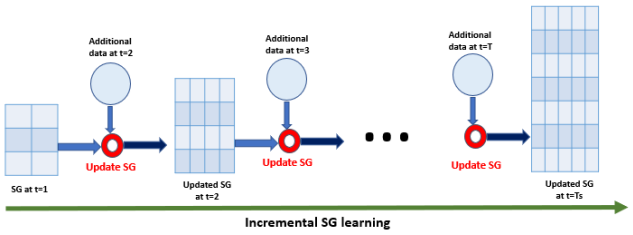


FIGURE 4: The incremental embedding model

2) Temporal Word Embeddings

In order to study the dynamics of the skip-gram model and track the movement of potential emerging keywords, we propose to learn word embeddings in a temporal fashion. To do so, we train the skip-gram model on the data resulting from the time-based preprocessing stage described in Section III-A2. Therefore, two training paradigms are proposed with respect to the generated corpora, namely *incremental embedding* for the incremental windows and *sliding embedding* for the sliding windows.

Incremental embedding. The incremental embedding goes through the corpora P to update word embeddings incrementally with the annual basis corpus augmentation. To do so, we propose two different embeddings. The first embedding aims to retrain the skip-gram model from scratch and perform a fresh model termed as *fresh embedding* in this paper. The second embedding, termed as *updated embedding*, reads the training data word by word to incrementally update the word frequency distribution and the noise distribution while performing stochastic gradient descent [54]. Figure 4 illustrates the incremental embedding model we use.

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

After selecting the first three corpora in the window starting from $t-1$ as mentioned in Section III-A2, the next corpora is selected from the window starting from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

t. The process is repeated iteratively until all P_t corpora are trained.

C. SIMILARITY COMPUTATION

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

$$similarity(w_i, w_j) = cosine(u_{w_i}, u_{w_j}) = \frac{u_{w_i} \cdot u_{w_j}}{\|u_{w_i}\| \|u_{w_j}\|} \tag{2}$$

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

D. POST-PROCESSING

After computing all similarity matrices corresponding to all T timespans, *Leap2Trend* proceeds with the ranking of the similarities of keywords in each matrix. The resulting ranked matrices are then used to identify the pairs of keywords having significant ascents in their ranking over time. These keywords are potentially considered as emerging trends due to their accelerating closeness. This step is termed as *rank ascent identification*.

1) Ranking

Given a similarity matrix $M^t_{i,j}$ of size $|v| \times |v|_{v \subseteq \mathscr{V}}$; that corresponds to the similarity values of a set v of keywords at a timespan t, our aim is to rank this matrix in order to define the set of closest pairs of keywords at this time period.

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

2) Rank Ascent Identification

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

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

Algorithm 1: Ranking Similarity Matrix

input : similarity matrix $M^t_{i,j}$
output: ranked ${M''_{i,j}}^t$

```
rank ← 0;
M'_{i,j}^t ← sort(M_{i,j}^t);
for i ← 1 to length(M'_{i,j}^t) do
    for j ← 1 to length(M'_{i,j}^t) do
        temp ← M'[i][j];
        for i' ← 1 to length(M'_{i,j}^t) do
            for j' ← 1 to length(M'_{i,j}^t) do
                if (M'[i'][j'] == temp) then
                    M''[i'][j'] ← rank + 1;
                    rank ← rank + 1;
                    j' ← length(M'_{i,j}^t);
                end
            end
        end
    end
end
```

$$\delta^{t,t+1} = M^{t,t+1}_{rank}[i][j] = M''[i][j]^t - M''[i][j]^{t+1} \tag{3}$$

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

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

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

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

Algorithm 2: Rank Ascent Identification

input : Ranked matrices $M''^{t}_{i,j}$, $M''^{t+1}_{i,j}$, threshold θ
output: fast emerging pairs of keywords$(w_i, w_j)^*$

for $t \leftarrow 1$ **to** T **do**
  for $i \leftarrow 1$ **to** $length(M''^{t}_{i,j})$ **do**
    for $j \leftarrow 1$ **to** $length(M''^{t}_{i,j})$ **do**
      for $i' \leftarrow 1$ **to** $length(M''^{t+1}_{i,j})$ **do**
        for $j' \leftarrow 1$ **to** $length(M''^{t+1}_{i,j})$ **do**
          $M_{rank}[i][j] \leftarrow M''^{t}[i][j] - M''^{t+1}[i'][j']$;
        end
      end
    end
  end
end
for $i \leftarrow 1$ **to** $length(M_{rank})$ **do**
  for $j \leftarrow 1$ **to** $T-1$ **do**
    $\delta = M_{rank}[i][j] - M_{rank}[i+1][j+1]$;
    if $(\delta > 0$ & $\delta \leq \theta)$ **then**
      return $(w_i, w_j)^*$;
    end
  end
end

IV. EXPERIMENTAL STUDY

In this section, we present an evaluation of *Leap2Trend* on the task of tracking and detecting emerging research trends. To this end, we selected two datasets related to two research areas: computer science and bioinformatics. Then, we evaluated the obtained results of our approach on two gold standards: Google Trends hits and Google Scholar citations.

A. DATASETS

The two corpora represent respectively 30 years of NIPS (Neural Information Processing Systems) conference papers and 15 years of MICCAI (Medical Image and Computer Assisted Intervention) conference papers.

1) NIPS Dataset

The NIPS corpora consist of the full text of 7241 papers published in Neural Information Processing Systems conference between 1987 and 2017. The dataset is publicly available on Kaggle[6] and contains information about papers, authors and the relation (papers-authors). We used the *papers* database that defines six features for each paper: id, title, event type (i.e., poster, oral or spotlight presentation), PDF name, abstract and paper text.

[6]https://www.kaggle.com/benhamner/nips-2015-papers/data

2) MICCAI Dataset

The MICCAI corpora consist of 15 years of Medical Image and Computer Assisted Intervention proceedings from 2004 to 2018 with a total of 3844 papers. MICCAI is one of the top conferences in the area of bioinformatics.
We crawled the proceedings from Springer website[7] under PDF format. Afterward, we extracted the text using the package *"pdftools"*[8] provided by R.

B. GOLD STANDARD

To evaluate the effectiveness of *Leap2Trend* in forecasting research trends, we need to find a set of trends determined a priori to be correct; known as gold standard. In the context of this study, we propose to use both *Google Trends*[9] and *Google Scholar* citations[10] as gold standards.

We choose Google Trends because it displays search trends data on Google; Google is considered the first place to start for researchers to find background on the research topic[11]. However, we use Google Scholar to collect the raw citations of publications.

1) Google Trends Hits

Google Trends analyses the popularity of search queries in Google Search[12] across various regions and languages and it compares the search volume of different queries over time[13].

Due to its ability to track various words and phrases that are typed into Google's search-box over time, we found that Google Trends aligns with *Leap2Trend* that tracks the closeness and the merge of pairs of scientific keywords over time towards new trends. To this end, we propose the following methodology to compare the results of *Leap2Trend* with Google Trends:

1) For each pair of keywords studied by *Leap2Trend*, we download the results from Google Trends that report the Google query volumes of this pair of keywords. Recall that the keywords are typed as they are in the interface of Google Trends without quotations for more than one-word keywords. We use the API *pytrends*[14] that downloads data in form of csv files recording the number of queries of this pair of keywords on a monthly basis. For convenience, we will refer to the number of queries of Google Trends as *Google Trends hits*. We set the parameter *'timeframe'* of *pytrends* to (2004-2017) and (2004-2018) for respectively NIPS and MICCAI corpora respecting the time-frame of both corpora as described in Section IV-A. The start date 2004 is justified by the start of Google Trends service. For this

[7]https://www.springer.com/
[8]https://cran.r-project.org/web/packages/pdftools/pdftools.pdf
[9]https://trends.google.com/
[10]https://scholar.google.com
[11]https://library.royalroads.ca/infoquest-tutorials/internet-searching/google-vs-google-scholar-which-one-do-i-use
[12]https://www.google.com/
[13]https://en.wikipedia.org/wiki/Google_Trends
[14]https://github.com/GeneralMills/pytrends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

reason, we ignore the NIPS set of publications before 2004 when we evaluate our approach. For the parameter *'geo'* that refers to the region of search, we set it to the by-default parameter which returns worldwide results. Recall that the retrieval time of Google Trends data is November and December 2018.

2) To be consistent to our results provided by *Leap2Trend* on yearly basis, we aggregate the Google Trends hits in the csv files by summing up the hits of each 12 months together.
3) Referring to Section III-D2, we define the ascents (jumps and leaps) in ranking over time of each pair of keywords. For each ascent, we track the Google Trends hits 3 years ahead and we compute the slope of the linear regression of these hits. Our aim behind this computation is to check if the jump in ranking captured by *Leap2Trend* indicates a positive slope and consequently defines this pair of keywords as emerging trend. This could show the predictive power of our approach in forecasting trends. The choice of 3 years as a duration is justified in Section III-A2 and the slope m_{hits} of the linear regression of Google Trends hits is defined as follows:

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

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

2) Google Scholar Citations

Google Scholar is the world's largest academic search engine in January 2018 with roughly 389 million documents indexed including articles, citations and patents [55].

Due to its ability to calculate and display the citation counts of scientific publications and its wide coverage of article published in English with an estimate of 100 million[15], we use Google Scholar to extract the raw citations of NIPS and MICCAI publications used in our evaluation. To do so, we use *Public or Perish* software[16] that uses Google Scholar to obtain the raw citations.

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

1) For each pairs of keywords studied by *Leap2Trend*, we select the set of all publications mentioning these keywords in their titles and we compute the total number of their citation counts returned by Google Scholar. We assume that the title plays a pivotal role in communicating research.
2) We compare the ascents of these keywords with the citation counts over timespans. A perfect result of *Leap2Trend* corresponds to a positive correlation between the ascents and the citations, i.e., when the jump increases, the citation count increases and vice versa.

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

C. EVALUATION METRICS

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

i How accurate is *Leap2Trend* in predicting future trends at an early stage?
ii How precise is *Leap2Trend* in following the flow of Google Trends hits and citation counts?

1) Ascent Accuracy and Recall

Ascent Accuracy. The ascent accuracy (accuracy) evaluates the prediction power of our approach in forecasting new trends at an early stage by tracking the ascents in ranking of pairs of keywords that will eventually lead to emerging trends. Therefore, we define the accuracy as the fraction of the number of $ascents^+$ – defined as the ascents that successfully led to positive slopes in the linear regression of Google Trends hits – with the number of ascents returned by *Leap2Trend* as expressed in Equation 5.

$$accuracy = \frac{|\{ascents^+\} \cap \{ascents\}|}{|\{ascents\}|} \tag{5}$$

Ascent Recall. The ascent recall (recall) attests the number of ascents in the gold standard that were successfully detected by our approach. Therefore, we define the recall as the fraction of the number of $ascents$ returned by our approach with the number of $ascents^+$ that successfully led to positive slopes in the linear regression of Google Trends hits, as expressed in Equation 6.

$$recall = \frac{|\{ascents^+\} \cap \{ascents\}|}{|\{ascents^+\}|} \tag{6}$$

2) Ascent Precision

We define two measures of ascent precision for our approach. The first measure P_{GT} evaluates the obtained results against Google Trends hits, while the second measure P_{GS} evaluates the results against Google Scholar citations. Both of them refer to how close are *Leap2Trend* ascents to Google Trends hits or citation counts. We mean by close how the ascents are positively correlated with the hits or the citations. Therefore, we formally define each of these two precision measures as the fraction of the number of $(w_i, w_j)^{corr+}$ – the pairs of keywords having positive correlations with the hits or citations – with the total number of pairs of keywords formed from the vocabulary v, $v \subseteq \mathcal{V}$ as defined in Section III-C. Equation 7 expresses the ascent precision (precision) where $precision$ refers to P_{GT} or P_{GS} with respect to the used gold standard.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

$$precision = \frac{|\{(w_i, w_j)^{corr+}\}|}{|\{(w_i, w_j)\}|}; i, j \in |v| \quad (7)$$

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

$$\rho = \frac{\sum_s (x_s - \bar{x})(y_s - \bar{y})}{\sqrt{\sum_s (x_s - \bar{x})^2 \sum_s (y_s - \bar{y})^2}} \quad (8)$$

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

D. RESULTS

For each of our datasets, namely NIPS and MICCAI, we ran three series of experiments within *Leap2Trend* framework following the three temporal embedding paradigms described in Section III-B2. Then, we evaluated the obtained results against the two gold standards Google Trends and citation counts defined in Section IV-B.

For both datasets, we started by selecting the keywords of interest to be studied in this research. We first started by selecting the top 100 frequent bigrams extracted from the titles of the publications. We chose bigrams rather than unigrams because of their frequent use in scientific corpora especially in computer science and bioinformatics; the two research areas we are studying in this paper. The selection of these keywords from the titles is justified by the fact that the title of a scientific paper is mostly self-explanatory reflecting the work being reported; hence it possibly contains the important keywords of interest in any research area. From these 100 bigrams, we kept only the bigrams whose their combination provides available information from Google Trends in order to fairly evaluate our approach against the gold standard. This restricted our keywords of interest to only 20 bigrams. This number has been also supported by *Google Hot Trends*[17] that displays the 20 hot and fastest rising search terms at a time. Similarly, our aim is to instantly detect the fastest rising trends in the field of study. The number of these emerging trends could not be high as we are tracking the evolution on yearly basis. For instance, a study performed by Hoonlor *et al.* [4] on evolving Computer Science research showed that the average length of the evolutionary chain is 4.5 years with few new topics. This has been also proved by a study conducted by Asooja *et al.* [56] on the domain of Natural Language Processing, Information Retrieval, and Semantic Web. They detected only two new topics in a period of 6 years from 2008 to 2014. Recall that we use the same set of pairs of keywords for all timespans in order to keep tracking their similarities/dissimilarities over time. We are aware that this approach prevents us to include new keywords/topics that may appear when time progresses. But, our main goal in to provide a proof of concept for our proposed approach *Leap2Trend* and proves its ability to detect emerging trends instantly.

After preprocessing both NIPS and MICCAI corpora, we trained the skip-gram model at every timespan with the embedding dimension $N = 200$ and the context window $= 6$. The choice of these hyperparameters is supported by our previous findings [53] that showed that these hyperparameters are optimal within scientific corpora. Recall that *Word2vec package* of the open source *Gensim Python Library*[18] has been used to implement the word vector representations. *Gensim* was ran on Windows Intel core i7 platform that supports *Python* and *NumPy*. For the incremental windows, we performed two trainings. The first training follows an updated embedding as described in Section III-B2 while the second training created a fresh trained model by retraining it from scratch. The code of these two trainings is publicly available here[19]. For the sliding windows, we trained the model at every timespan, because the sliding paradigm results in new vocabulary forgetting one year vocabulary and adding one year ahead vocabulary as shown in Fig. 3.

After each training at a timespan t, we created a similarity matrix $M^t_{i,j}$ as described in Section III-C that corresponds to the 20 keywords of interest extracted from the titles of the publications as described above.

At every timespan t, we ranked the similarity values of $M^t_{i,j}$ and then we created $M^{t,t+1}_{rank_{i,j}}$ that stores the difference in ranking of the pairs of keywords between two subsequent timespans t and $(t+1)$. For each pair of keywords, we picked all ascents corresponding to a positive δ calculated following Equation 3.

1) Leap2Trend vs Google Trends hits

For each ascent, we computed the slope of the linear regression of Google Trends hits as expressed in Equation 4. In order to avoid bias, we ignored the ascent picked at 2005, because it corresponds to the ascent in ranking of the pair of keywords between 2004 and 2005 while δ at 2004 is set to 0 (2004 is the starting year of analysis and corresponds to the starting year of Google Trends). After the selection of all ascents related to all studied pairs of keywords, we computed the related accuracy as described in Equation 5. This accuracy corresponds to any ascent. Then, we set different thresholds for δ: $\{5, 10, 20, 30\}$ defining leaps with various magnitudes. The choice of these thresholds was based on the overall obtained values of δ on both datasets after the three training paradigms. For this reason, we may not find some of these thresholds on some results such as the thresholds 20 and 30 in the *fresh embedding* of MICCAI as shown in Fig. 6.

Fig. 5 and Fig. 6 show accuracy measures of *Leap2Trend* with the three embedding paradigms: *fresh embedding*, *up-*

[17] https://en.wikipedia.org/wiki/Google_Trends

[18] https://radimrehurek.com/gensim/models/word2vec.html

[19] https://github.com/AmnaKRDB/Leap2Trend

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

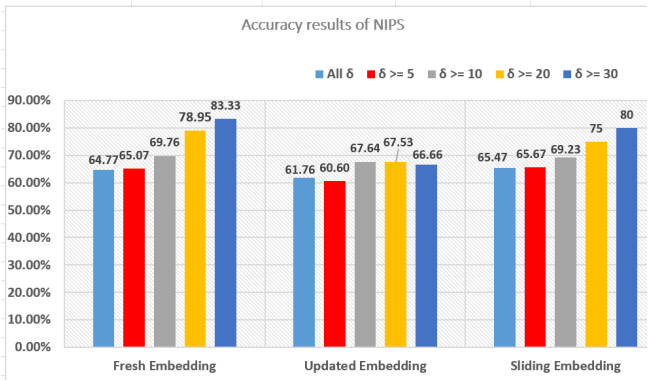


FIGURE 5: Accuracy results of NIPS with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

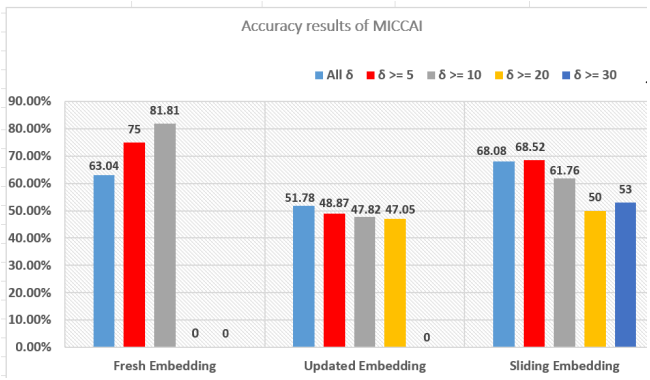


FIGURE 6: Accuracy results of MICCAI with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

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

The overall results shown in Fig. 5 and Fig. 6 reveal that the best accuracy is given when experimenting (i) with the fresh embedding and (ii) with high leaps. (i) could be justified by the fact that the ideal approach for incremental embedding would be to retrain the model from scratch including new vocabulary in the training corpus [54]. That is because the incremental training of word embeddings may drift words learned from later batches arbitrary far from words in earlier batches that are not re-presented. This observation is supported by our results on both NIPS and MICCAI where the updated embedding performed the worst in all settings. (ii) highlights the importance of the magnitude of ascents; when the ascent increases, the accuracy increases accordingly. The substantial improvement in accuracy from any δ to a greater

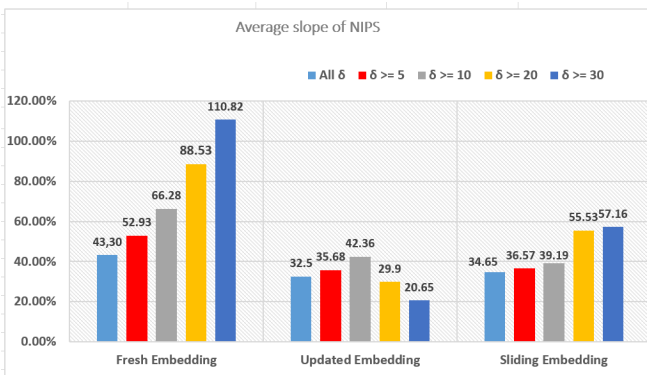


FIGURE 7: Average slope results of NIPS with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

threshold underlines the ability of *Leap2Trend* to accurately forecast trends instantly by paying attention to the ascents in ranking of pair of keywords over time.

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

$$\Delta m_{hits} = \frac{\sum_{i=1}^{|\{ascents\}|} m_{hits}}{|\{ascents\}|} \tag{9}$$

Fig. 7 and Fig. 8 illustrate the obtained results of average slopes on NIPS and MICCAI respectively. Similar to the previous results of accuracy, the fresh embedding performs the best in both datasets. For instance, the average of slopes Δm_{hits} gradually increases with the increase of ascents. However, for the sliding embedding related to MICCAI dataset, we noticed a decrease in Δm_{hits} starting from the threshold $\delta \geq 10$. This is justified by the rarity of picked ascents with higher magnitude. As a matter of fact, this decrease goes in parallel with the accuracy that drops to 50% with $\delta \geq 20$ as shown in Fig. 6. In reality, this 50% represents 4 positive slopes over 8 detected ascents with more than 20 ascents. Therefore, both the average of slopes and the accuracy are highly sensitive to the magnitude of ascents.

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access

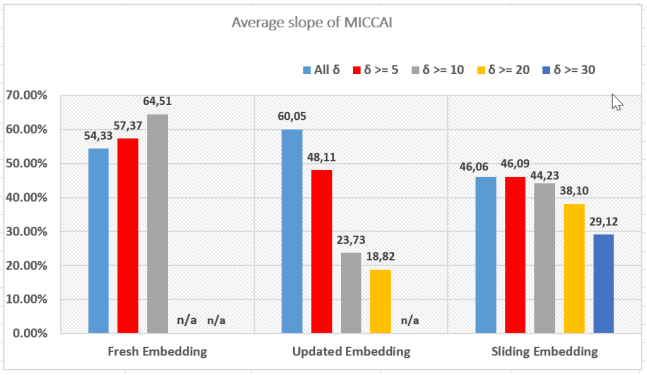


FIGURE 8: Average slope results of MICCAI with respect to the three embedding paradigms and different thresholds of δ, $\delta > 0$ in all cases

As a proof of evidence, *Leap2Trend*, applied to NIPS with the fresh embedding, detects an ascent of $\delta = 11$ of the pair of keywords (*"neural_network – machine_learning"*) between 2012 and 2013 as shown in Fig. 10a. This ascent is highly significant as this pair of keywords reflects the emerging research trend *deep learning* defined as neural-based learning. This emerging research trend started to flourish as independent research topic in 2014. This is insightful as it shows that *Leap2Trend* was able to predict the merge of these two keywords toward a new scientific keyword. In fact, the average of slope Δm_{hits} which is equal to 76.9 supports this assumption. The strength of this slope is clearly shown in Fig. 10a as the number of Google Trends hits passes from 86 in the year when the ascent happened to 329 after 3 years. Similarly, *Leap2Trend*, applied to MICCAI with the fresh embedding, detects an ascent of $\delta = 10$ of the pair of keywords (*"lung_cancer – breast_cancer"*) between 2009 and 2010. This ascent was insightful as the statistics on medical research in 2010 showed that *lung cancer* was the most second commonly diagnosed cancer in the UK after *breast cancer*[20]. This could justify why *Leap2Trend* detected the ascent of these two keywords as they co-occur together. This observation is then supported by the average of slope Δm_{hits} which is equal to 40.1 and shows an increase in Google Trends hits in Fig. 8.

Overall, the accuracy results on NIPS and MICCAI show a great potential of our approach to predict research trends instantly. It is also important to reveal how many of the ascents presented in the gold data were detected by *Leap2Trend*. To do so, we computed the recall as defined in Equation 6. A relevant ascent, named as $ascent^{+}$, is defined as an ascent approved by a positive slope of the Google Trends hits looking three years ahead. Fig. 9 shows recall measures of *Leap2Trend* with the three embedding paradigms applied to NIPS and MICCAI datasets. The overall results show promising findings in recalling research trends instantly. For instance, the recall is above 50% in all settings on NIPS, and it reaches and exceeds 40% on MICCAI. *Leap2Trend* reveals then a great potential to recall trends ahead in time. The obtained recall results on both datasets align with the accuracy results on the impact of every embedding setting. As a matter of fact, the fresh embedding performs the best with NIPS dataset with 57.79% while the sliding embedding performs the best with MICCAI dataset with 43.83% for the same reasons detailed for the accuracy. Exceptionally for recall with MICCAI, we found that the two embedding settings (fresh and updated) perform similarly with 39.72%. This could be justified by the size of corpora as MICCAI has small corpora with more likely few new keywords which makes the incremental embedding less sensitive to the followed paradigm whether it is fresh or updated.

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

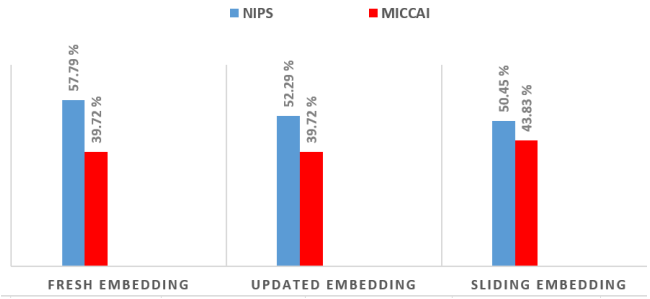
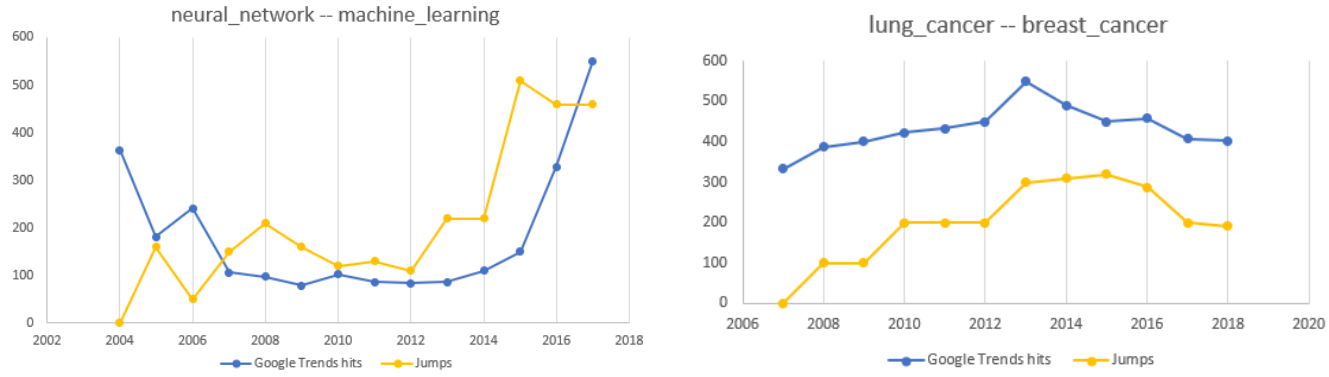


FIGURE 9: NIPS and MICCAI recall against Google Trends hits

After testing the effectiveness of *Leap2Trend* in predicting research trends instantly using accuracy, we tested the closeness of *Leap2Trend* ascents to Google Trends hits by performing a fine-grained analysis. This fine-grained analysis aims to check to what extent the ascents of *Leap2Trend* are correlated with Google Trends hits. To do so, we computed the Spearman's correlation coefficient (Equation 8) related to every pair of keywords. Afterward, we measured the precision P_{GT} following Equation 7. Fig. 11 illustrates the obtained precision results on NIPS and MICCAI with the three embedding paradigms.

Interestingly, these results indicate that the sliding embedding in both datasets performs significantly better than the incremental embedding (the fresh embedding and the updated embedding) with a precision of 88.88% and 61.53% for NIPS and MICCAI respectively. This could be justified by the fact that the sliding window of 3 years length could perfectly match the keywords published in the papers with the keywords used in Google Search unlike the incremental window that keeps the old vocabulary. This affects the similarity of keywords and consequently affects their ranking and hence their ascents. For the updated embedding, the precision results confirm those previously obtained with accuracy; it performs worst with all measures of effectiveness. Overall, the P_{GT} results support the accuracy ones and show that our approach *Leap2Trend* is able to forecast trends instantly matching Google Trends hits. For instance, the Spearman's correlation coefficient shows a strong correlation between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2957440, IEEE Access



(a) NIPS Example

(b) MICCAI Example

FIGURE 10: The linear regression of jumps and Google Trends hits related to two pairs of keywords from NIPS and MICCAI[a=] [a=]For visualisation purpose, the values of jumps were multiplied by 10 in order to clearly display the jumps with respect to the Google Trends hits

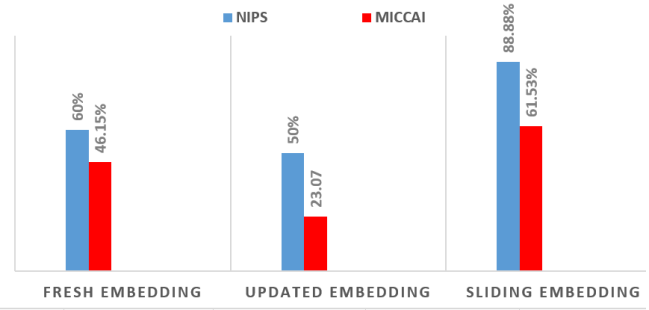


FIGURE 11: NIPS and MICCAI precision against Google Trends hits

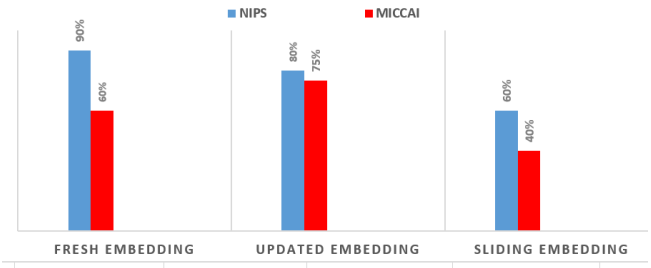


FIGURE 12: NIPS and MICCAI precision against Google Scholar citations

Leap2Trend ascents and Google Trends hits for the sliding embedding with 65% and 55% of ρ values greater than 0.6 for respectively NIPS and MICCAI datasets.

For all settings and measures, NIPS dataset is performing much better than MICCAI dataset. This could be justified by two reasons. The first reason refers to the size of corpora; NIPS corpora is much bigger than MICCAI corpora and it has been proved in the literature [17], [19] that word embedding quality increases as the corpus size increases. The second reason may refer to the popularity and the strength of the conference. For instance, NIPS conference is more than 30 years old while MICCAI is only 21 years old. The prestige of the conference contributes to its strength and rapidity in developing new research topics.

2) Leap2Trend vs Google Scholar citations

In order to support the findings of *Leap2Trend* obtained against the gold standard Google Trends hits, we performed a new validation of *Leap2Trend* results with the citation counting approach which is widely used in the literature and provides a snapshot of a fast-growing field. To do so, we retrieved academic citations from Google Scholar of all the NIPS and MICCAI publications as described in Section IV-B2. Then, we compared the ascents of all studied pairs of keywords from NIPS and MICCAI over the three embedding paradigms with citation counts.

Similar to our results performed with Google Trends hits, we used Spearman's correlation coefficient (Equation 8) to measure the correlation between the ascents and the citation counts. Afterward, we computed the precision P_{GS} of *Leap2Trend* results on NIPS and MICCAI following Equation 7.

Fig. 12 demonstrates the obtained precision results on NIPS and MICCAI with the three embedding paradigms. According to these results, the two incremental embeddings outperform the sliding embedding in both datasets with a precision that reaches 90% with the fresh embedding applied to NIPS dataset. These results are meaningful because the incremental embedding keeps the history of publications which affects the similarity of keywords and consequently affects their ascents. This perfectly matches the citation counting approach that takes time to progress and reveal trends. However, the sliding embedding refers to only 3 years publications with a forgotten one year publications and an added new one year publications. This window size is not enough to reflect the citation counts that need time to evolve.

Overall, *Leap2Trend* precision results against Google

Scholar citations support the previous results on Google Trends hits as well as accuracy, and they show the effectiveness of our approach to detect emerging trends with promising findings.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose *Leap2Trend*, a new approach to instant detection of research trends, and we show how it performs in the research areas of computer science and bioinformatics.

This work addressed this challenge in an innovative way by harnessing word embedding techniques to drill into the paper content and track the dynamics of similarities between pairs of keywords. To do so, our *Leap2Trend* approach trained temporal embeddings following two temporal paradigms: incremental and sliding. Then, after each training, it created a similarity matrix that stores the similarities of pairs of keywords of interest. Afterward, it ranked the entries of this matrix and computed the ascents in ranking over different timespans. Finally, for each picked ascent, *Leap2Trend* performed different evaluations against *Google Trends hits* and *Google Scholar citations* in order to test if the detected ascent of the pair of keywords refers to a new emerging topic. The obtained results showed the effectiveness of our approach to detect emerging keywords instantly.

While these results are satisfactory, our research presents some limitations that we plan to address in future work. First, the gold standard *Google Trends* (i) does not cover the years before 2004, and (ii) only provides a relative search value and does not provide an exact search volume. We thus intend to consider more big scholarly data resources to fill in this gap and provide a more robust gold standard that could be adopted to further study the task of detecting emerging scientific trends. Second, the current version of *Leap2Trend* only focuses on hits in ranking to study the dynamics of research topics. This indicator may not be enough to fully understand the dynamics of science. We thus plan to investigate *falls* in ranking and study their impact to show the outdated research topics. Finally, *Leap2Trend* has been only tested on the fields of computer science and bioinformatics. We believe that more work is needed to evaluate it on other disciplines such as *physics* or *biology*.

In the future, we also plan to investigate different resources that exist outside the realm of research papers, such as online media and social networks to detect emerging trends. This is motivated by the fact that scientists and researchers are increasingly using social media to discover new research opportunities, discuss research with colleagues and disseminate research information which allows to track public attention and public recognition of emerging topics.

REFERENCES

[1] Z. Wu, J. Wu, M. Khabsa, K. Williams, H. Chen, W. Huang, S. Tuarob, S. R. Choudhury, A. Ororbia, P. Mitra, and C. L. Giles, "Towards building a scholarly big data platform: Challenges, lessons and opportunities," in IEEE/ACM Joint Conference on Digital Libraries, 2014, pp. 117–126.

[2] Y. An, M. Han, and Y. Park, "Identifying dynamic knowledge flow patterns of business method patents with a hidden markov model," Scientometrics, vol. 113, no. 2, pp. 783–802, Nov 2017.

[3] S. Effendy and R. H. Yap, "Analysing trends in computer science research: A preliminary study using the microsoft academic graph," in Proc. of the 26th International Conference on World Wide Web Companion, 2017, pp. 1245–1250.

[4] A. Hoonlor, B. K. Szymanski, and M. J. Zaki, "Trends in computer science research," Commun. ACM, vol. 56, no. 10, pp. 74–83, Oct. 2013.

[5] J. Hou, X. Yang, and C. Chen, "Emerging trends and new developments in information science: a document co-citation analysis (2009–2016)," Scientometrics, vol. 115, no. 2, pp. 869–892, May 2018.

[6] D. E. Rossetto, R. C. Bernardes, F. M. Borini, and C. C. Gattaz, "Structure and evolution of innovation research in the last 60 years: review and future trends in the field of business through the citations and co-citations analysis," Scientometrics, Mar 2018.

[7] A. A. Salatino, F. Osborne, and E. Motta, "AUGUR: forecasting the emergence of new research topics," in Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018, 2018, pp. 303–312.

[8] A. Santa Soriano, C. Lorenzo Álvarez, and R. M. Torres Valdés, "Bibliometric analysis to identify an emerging research area: Public relations intelligence—a challenge to strengthen technological observatories in the network society," Scientometrics, Feb 2018.

[9] C. Weismayer and I. Pezenka, "Identifying emerging research fields: a longitudinal latent semantic keyword analysis," Scientometrics, vol. 113, no. 3, pp. 1757–1785, Dec 2017.

[10] C. Zhang and J. Guan, "How to identify metaknowledge trends and features in a certain research field? evidences from innovation and entrepreneurial ecosystem," Scientometrics, vol. 113, no. 2, pp. 1177–1197, Nov 2017.

[11] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big scholarly data: A survey." IEEE Trans. Big Data, vol. 3, no. 1, pp. 18–35, 2017.

[12] Z. Taskin and U. Al, "A content-based citation analysis study based on text categorization," Scientometrics, vol. 114, no. 1, pp. 335–357, 2018.

[13] A. Anderson, D. McFarland, and D. Jurafsky, "Towards a computational history of the acl: 1980-2008," in Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, 2012, pp. 13–21.

[14] A. Bakarov, A. Kutuzov, and I. Nikishina, "Russian computational linguistics: Topical structure in 2007-2017 conference papers," in Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2018", 06 2018.

[15] C. Li, S. Feng, Q. Zeng, W. Ni, H. Zhao, and H. Duan, "Mining dynamics of research topics based on the combined lda and wordnet," IEEE Access, vol. 7, pp. 6386–6399, 2019.

[16] M. Paul and R. Girju, "Topic modeling of research fields: An interdisciplinary perspective," International Conference Recent Advances in Natural Language Processing, RANLP, pp. 337–342, 2009.

[17] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in NAACL-HLT, 2013, pp. 746–751.

[18] A. Kutuzov, L. Øvrelid, T. Szymanski, and E. Velldal, "Diachronic word embeddings and semantic shifts: a survey," in Proc. of the 27th International Conference on Computational Linguistics, 2018, pp. 1384–1397.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 3111–3119.

[20] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific paper recommendation: A survey," IEEE Access, vol. 7, pp. 9324–9339, 2019.

[21] X. Bai, "Predicting the number of publications for scholarly networks," IEEE Access, vol. 6, pp. 11 842–11 848, 2018.

[22] 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.

[23] J. Liu, T. Tang, W. Wang, B. Xu, X. Kong, and F. Xia, "A survey of scholarly data visualization," IEEE Access, vol. 6, pp. 19 205–19 221, 2018.

[24] J. Zhang, B. Xu, J. Liu, A. Tolba, Z. Al-makhadmeh, and F. Xia, "Pepsi: Personalized prediction of scholars' impact in heterogeneous temporal academic networks," IEEE Access, vol. 6, pp. 55 661–55 672, 2018.

[25] M. Shardlow, R. Batista-Navarro, P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Identification of research hypotheses and new knowl-

edge from scientific literature," BMC Medical Informatics and Decision Making, vol. 18, no. 1, 2018.

[26] J. Kim, D. Kim, and A. Oh, "Joint modeling of topics, citations, and topical authority in academic corpora," Transactions of the Association for Computational Linguistics, vol. 5, pp. 191–204, 2017.

[27] K. W. Boyack, C. Smith, and R. Klavans, "Toward predicting research proposal success," Scientometrics, vol. 114, no. 2, pp. 449–461, Feb 2018.

[28] Y. Liu, Z. Huang, Y. Yan, and Y. Chen, "Science navigation map: An interactive data mining tool for literature analysis," in Proceedings of the 24th International Conference on World Wide Web, ser. WWW '15 Companion, 2015, pp. 591–596.

[29] M. M. Alam and M. A. Ismail, "Rtrs: a recommender system for academic researchers," Scientometrics, vol. 113, no. 3, pp. 1325–1348, Dec 2017.

[30] R. Dey, A. Roy, T. Chakraborty, and S. Ghosh, "Sleeping beauties in computer science: characterization and early identification," Scientometrics, vol. 113, no. 3, pp. 1645–1663, Dec 2017.

[31] S. Effendy, I. Jahja, and R. H. Yap, "Relatedness measures between conferences in computer science: A preliminary study based on dblp," in Proceedings of the 23rd International Conference on World Wide Web, ser. WWW '14 Companion, 2014, pp. 1215–1220.

[32] S. Effendy and R. H. C. Yap, "The problem of categorizing conferences in computer science," in Research and Advanced Technology for Digital Libraries, N. Fuhr, L. Kovács, T. Risse, and W. Nejdl, Eds., 2016, pp. 447–450.

[33] S. Kim, D. Hansen, and R. Helps, "Computing research in the academy: insights from theses and dissertations," Scientometrics, vol. 114, no. 1, pp. 135–158, Jan 2018.

[34] L. Bornmann and H. Daniel, "What do citation counts measure? a review of studies on citing behavior," Journal of Documentation, vol. 64, no. 1, pp. 45–80, 2008.

[35] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in Proc. of Conference on Empirical Methods in Natural Language Processing, EMNLP'2016, 2016, pp. 595–605.

[36] M. Huang, Q. Qian, and X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," ACM Trans. Inf. Syst., vol. 35, no. 3, pp. 26:1–26:27, Jun. 2017.

[37] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Enhancing topic modeling for short texts with auxiliary word embeddings," ACM Trans. Inf. Syst., vol. 36, no. 2, pp. 11:1–11:30, Aug. 2017.

[38] G. D. Rosin, E. Adar, and K. Radinsky, "Learning word relatedness over time," in Proc.of Conference on Empirical Methods in Natural Language Processing, EMNLP'2017, 2017, pp. 1168–1178.

[39] V. D. Carlo, F. Bianchi, and M. Palmonari, "Training temporal word embeddings with a compass," in The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, 2019, pp. 6326–6334.

[40] Y. Kim, Y. Chiu, K. Hanaki, D. Hegde, and S. Petrov, "Temporal analysis of language through neural language models," in LTCSS@ACL, 2014, pp. 61–65.

[41] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena, "Statistically significant detection of linguistic change," in Proc. of the 24th International Conference on World Wide Web, 2015, pp. 625–635.

[42] T. Szymanski, "Temporal word analogies: Identifying lexical replacement with diachronic word embeddings," in Proc. of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 448–453.

[43] Y. Zhang, A. Jatowt, S. S. Bhowmick, and K. Tanaka, "The past is not a foreign country: Detecting semantically similar terms across time," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 10, pp. 2793–2807, Oct 2016.

[44] H. Dubossarsky, S. Hengchen, N. Tahmasebi, and D. Schlechtweg, "Time-out: Temporal referencing for robust modeling of lexical semantic change," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Jul. 2019, pp. 457–470.

[45] R. Mihalcea and V. Nastase, "Word epoch disambiguation: Finding how words change over time," in Proc. of the 50th Annual Meeting of the Association for Computational Linguistics, 2012, pp. 259–263.

[46] W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," in ACL (1), 2016.

[47] A. Kutuzov, E. Velldal, and L. Øvrelid, "Tracing armed conflicts with diachronic word embedding models," in Proc. of the Events and Stories in the News Workshop, 2017, pp. 31–36.

[48] Z. Yao, Y. Sun, W. Ding, N. Rao, and H. Xiong, "Dynamic word embeddings for evolving semantic discovery," in Proc. the 11th ACM International Conference on Web Search and Data Mining, 2018, pp. 673–681.

[49] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," CoRR, vol. abs/1301.3781, 2013.

[50] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in EMNLP, vol. 14, 2014, pp. 1532–1543.

[51] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[52] B. Wang, A. Wang, F. Chen, Y. Wang, and C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," CoRR, vol. abs/1901.09785, 2019.

[53] A. Dridi, M. M. Gaber, R. M. A. Azad, and J. Bhogal, "k-nn embedding stability for word2vec hyper-parametrisation in scientific text," in Discovery Science, 2018, pp. 328–343.

[54] N. Kaji and H. Kobayashi, "Incremental skip-gram model with negative sampling," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2017, pp. 363–371.

[55] M. Gusenbauer, "Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases," Scientometrics, Nov 2018.

[56] K. Asooja, G. Bordea, G. Vulcu, and P. Buitelaar, "Forecasting emerging trends from scientific literature," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), May 2016, pp. 417–420.

AMNA DRIDI received the B.Sc. degree in informatics and multimedia from Gabes University, Tunisia, in 2008, the B.Sc. degree (Hons.) in computer science from Sousse University, Tunisia, in 2010 and the M.Sc. degree in business informatics from Tunis University, Tunisia, in 2012. She is currently a Ph.D. Researcher with the School of Computing and Digital Technology, Birmingham City University. From 2014 to 2016, she was a Research Assistant with two Italian universities, respectively Free University of Bozen-Bolzano and the University of Cagliari. Her research interest includes information retrieval, social media analysis, sentiment analysis and natural language processing. She has published a number of peer-reviewed scientific articles in international conferences and journals. Amna's awards include a full Ph.D scholarship at Birmingham City University (UK), VLDB Travel fellowship in 2013, ECIR travel grant in 2016 and studentships in Italy.

MOHAMED MEDHAT GABER is a Professor in Data Analytics at the School of Computing and Digital Technology, Birmingham City University. Mohamed received his PhD from Monash University, Australia. He then held appointments with the University of Sydney, CSIRO, and Monash University, all in Australia. Prior to joining Birmingham City University, Mohamed worked for the Robert Gordon University as a Reader in Computer Science and at the University of Portsmouth as a Senior Lecturer in Computer Science, both in the UK. He has published over 200 papers, co-authored 3 monograph-style books, and edited/co-edited 6 books on data mining and knowledge discovery. His work has attracted well over four thousand citations, with an h-index of 35. Mohamed has served in the program committees of major conferences related to data mining, including ICDM, PAKDD, ECML/PKDD and ICML. He has also co-chaired numerous scientific events on various data mining topics. Professor Gaber is recognised as a Fellow of the British Higher Education Academy (HEA). He is also a member of the International Panel of Expert Advisers for the Australasian Data Mining Conferences. In 2007, he was awarded the CSIRO teamwork award.

R. MUHAMMAD ATIF AZAD is a Senior Fellow of Higher Education Academy. He specialises in the subject matter of Computer Science, Machine Learning, Evolutionary Computing (Genetic Programming, Genetic Algorithms, and Grammatical Evolution), Data Analytics, and Statistics. He received his PhD from Biocomputing and Developmental Systems Group at University of Limerick (UL). Since the year 2000, he has extensively worked on theory and applications of Machine Learning, and Nature Inspired Machine Learning (Evolutionary Computing), and has conducted internationally acclaimed work winning awards and honours from recognized international scientific fora. He has received the Best Reviewer Award at the European Conference on Genetic Programming (EuroGP) 2015, Copenhagen, Denmark, and Silver HUMIES Award at Genetic and Evolutionary Computation Conference (GECCO) 2015 for his work on Automatic Parallel Programming. He has published a number of peer-reviewed scientific articles in international conferences and journals. His research interests include Evolutionary Algorithms Genetic Programming, Genetic Algorithms, Grammatical Evolution, Automatic Programming, Machine Learning and Data Analysis.

JAGDEV BHOGAL received the B.Sc. degree (Hons.) in computer science in 1986, the MPhil degree in Expert Systems and Database Administration in 1989, and the Ph.D. degree in Computer Science in 2011. She is currently the Course Leader for MSc Business Intelligence and the MSc Big Data Analytics courses at Birmingham City University. She has published conference and journal papers on relational/object/nosql database systems, ontologies, text mining and cloud computing. She has professional certification: Oracle Certified Associate - Oracle Database 11g Administrator.