Identifying Emergent Research Trends by Key Authors and Phrases

Shenhao Jiang[†], **Animesh Prasad**[†], **Min-Yen Kan**[†], **Kazunari Sugiyama**[†‡]
[†]School of Computing, National University of Singapore
shenhao_jiang@u.nus.edu
{animesh|kanmy|sugiyama}@comp.nus.edu.sg
[‡]National Institute of Informatics, Japan
zakugus@nii.ac.jp

Abstract

Identifying emergent research trends is a key issue for both primary researchers as well as secondary research managers. Such processes can uncover the historical development of an area, and yield insight on developing topics. We propose an embedded trend detection framework for this task which incorporates our bijunctive hypothesis that important phrases are written by important authors within a field and vice versa. By ranking both author and phrase information in a multigraph, our method jointly determines key phrases and authoritative authors. We represent this intermediate output as phrasal embeddings, and feed this to a recurrent neural network (RNN) to compute trend scores that identify research trends. Over two large datasets of scientific articles, we demonstrate that our approach successfully detects past trends from the field, outperforming baselines based solely on text centrality or citation.

1 Introduction

At no time prior to now has the advance of science – as measured by the rate of growth in scientific publications – bloomed more rapidly. Even practitioners of narrowly-defined, specific fields need help to scan the masses of literature to summarize work and to identify developments likely to spur long-term impact due to the potential problem of information excess (Fujita et al., 2012). The identification of such **emergent research trends** — "radically novel and relatively fast growing research topic characterized by a certain degree of coherence, and a considerable scientific impact" (Wang, 2017) — is a task that is growing in importance and urgency with the increasingly large scale of scholarly literature.

Wang (2017)'s survey reveals two important techniques that characterize the state-of-the-art work on this task. First, text mining — specifically in the guise of topic models — have been employed to yield a summative form of large corpora. Both the Dynamic Topic Model (Blei and Lafferty, 2006) and Author–Topic Model (Rosen-Zvi et al., 2004) cater towards different aspects of the special temporal and authoring aspects of topics in scientific documents, specializing the basic Latent Dirichlet allocation (LDA; Blei et al. (2003)) topic model. Second, dense citations among papers has been used to measure the coherence contributing to a well-defined research trend (Price, 1965). Through appropriate metrics, core works can be identified, from which extracted keywords are treated as a trend's central concepts. Analyses often establish a co-citation network, as typified by Shibata et al. (2008).

These works point to the importance of side information external from the content itself, such as publication date or author identity, and its necessary incorporation into a detection methodology. Our observation is that these attributes often correlate with each other, naturally hinting to a framework where such information can reinforce their signal.

We thus take inspiration from other natural language processing and information retrieval scenarios where such mutual recursion has found application. Hyperlink Induced Topic Search (HITS; (Kleinberg, 1999)) showed web pages arranged as nodes of a hyperlinked graph can be mutually related and ranked by an authority and a hub (popularity) score, where the two scores are calculated using mutual recursion.

Proceedings of the 27th International Conference on Computational Linguistics, pages 259–269
Santa Fe, New Mexico, USA, August 20-26, 2018.

Yan et al. (2017) applied a similar logic to the task of ranking sentences and words with mutual recursion, where it is assumed that important sentences and important words are correlated. They achieved better performance for keyphrase extraction compared with TextRank (Mihalcea and Tarau, 2004), a non-recursive formulation.

Furthermore, with the current advent of deep learning and better semantic representations via embeddings, there is opportunity to utilize both advances in trend detection.

Applying these observations to the research trend task, we make the following assumptions when formulating our model detailed in Section 3:

- Important authors often collaborate together, *i.e.*, the co-authorship among key-authors contributes to the research trend emergence.
- Important authors often write important keyphrases. Such phrases are then often trending phrases for a research area.
- High centrality values for nodes are key features to identify the core items within a graph. For identifying emergent trends, these metrics must exhibit values that connote both importance and urgency.
- We neuralize the traditional trend detection pipeline. Specifically, we represent obtained keyphrases as averaged *tf-idf* word embeddings and use a recurrent neural network (RNN) model to perform supervised learning for trend status.

We use a multigraph ranking (MGR) core component to compute trending scores for keyphrases. We take these MGR outputs as input to our neuralized embedded trend detection (EDM) framework: namely, we represent the outputs as word or phrase embeddings, and feed their ranking scores into a recursive neural network to identify trends per time slice in a target dataset. This results in our final embedded trend detection model – ETD (MGR), or "ETD-M" for short.

2 Prior Work

Previous studies have addressed both of our tasks of obtaining keyphrases and identifying research fronts from scientific publications. We review this area first, then review mutually recursive algorithms used in practice.

Research Trend Detection. Wang (2017) classified research trend detection models into two categories, namely, text- and bibliometric-based approaches. In text-based approaches, keywords and terms representing the core topics have been the focus of study. Hall et al. (2008) performed post-hoc analysis by applying the LDA model to the batch of publications in the same year; top keywords generated in different years are compared in an effort to identify the transition of topics. Mörchen et al. (2008) attempts to combine keywords and the time of their appearances: the variation of term frequencies in two different timestamps provides the insights on how significant the term is becoming. Sessions within conferences are another perspective that have been investigated. Furukawa et al. (2014) constructed *tf-idf* vectors (Salton et al., 1983) from publications' abstracts to obtain the relative importance of sessions, and then applied the vectors to assess the evolution of a publication venue.

In bibliometric approaches, citations are used to connect papers. Hopcroft et al. (2004) have proposed the concept of co-citation, where papers are connected if they are both cited by another paper published later; following the co-citation relation, the graph of papers can be used for further studies (Shibata et al., 2008): topological clusters inside the graph was identified, where tightly knit clusters with densely distributed edges represent the general topics. Subsequently, NLP techniques such as *tf-idf* are employed to generate the top terms associated with the topic. Similar to textual approaches, top keywords from clusters in different years can then be used for research fronts analysis. Elkiss et al. (2008) tried to analyze the cohesion of texts based on the citing sentences and co-citation metrics, which provides insights on the

focused perspective of the cited paper. He et al. (2009) proposed an inheritance topic model to identify topic evolution by fully utilizing the impact of citations.

We note that both textual and bibliometric approaches expose shortcomings: they are restricted to historical data, and human interpretations are often required to establish the link between different timestamps; additionally, without a comprehensive keywords extraction method, the final output can be ambiguous and uninformative.

Mutual Recursive Algorithms. Hyperlink Induced Topic Search (HITS) Algorithm (Kleinberg, 1999) is a prototypical example and used for ranking pages on the Internet, where each page is assigned with two values: an "authority" score and a "hub/popularity" score. The rationale states that a good hub page points to many other pages, where a good authority page is pointed to by many hubs. Therefore, the two scores are computed dependently on each other.

In natural language processing, the implication of mutual recursion can apply to keyphrase extraction models: Yan et al. (2017) implemented a ranking model by constructing separate sentence and word graphs. They assume that important sentences and words are correlated, hence by linking the two graphs together and ranking sentences and words recursively, salient keywords can be extracted. Their model outperforms standard TextRank algorithm (Mihalcea and Tarau, 2004). Zhang et al. (2017) also proposed a graph-based keyphrase extraction approach by capturing mutual influences among different types of information on a candidate word, which have been widely used in existing supervised or graph-based approaches.

3 Methodology

Our goal is to mine and assign high fidelity importance scores to keyphrases, and then to use these to predict emerging trends, based on the abstracts for a large target scientific publication dataset. Our embedded trend detection (ETD) model consists of the following three components: 1) extraction of keyphrases, 2) representativeness enhancement with word embeddings, and 3) predictions and rankings of phrase with RNN. In the following, we detail each of the components.

3.1 Multigraph Ranking for Key Phrase and Author Extraction

Our ETD framework admits any means of identifying the keyphrases, so any keyphrase generation algorithm can be employed (*e.g.*, TextRank). However, since our task is slightly different in that we want to generate keyphrases for overall trend detection of an area (as opposed to the more typical characterization of a single publication), we need to introduce several refinements.

As a result, we propose our multigraph ranking (MGR) component for salient keyphrase extraction. Our MGR model starts with the extraction of keyphrases and key-authors based on the assumption that important authors are more likely to write potentially trending phrases. We extract them via iterative score ranking in a pair of jointly-connected graphs.

We construct the graphs as follows: documents are first grouped into year-based clusters. For each of the cluster, we build an "author" graph, which contains individual authors who have published at least one paper in that year; authors are connected by undirected edges whose weights are determined by the number of publications two authors have made together during the year. We also construct a "phrase" graph following the logic of TextRank with modifications. Compare this to the standard TextRank model, where the nodes in the graph are tokenized words from the raw text, and the edges conceptualize the adjacency when tokens occur within a predefined window size.

Our empirical results demonstrate that domain-specific single-word terms dominate the final keyphrase list. Hence in our model, we pre-process the input to extract noun phrases via regular expression over parts-of-speech[1] before TextRank processing.

Consider the sample document in Figure 1: after transforming the raw text to a set of noun phrases, each noun phrase is split further: for {coherency protocols}, two separate nodes will be added to the

[1]Using the Natural Language Toolkit (NLTK) via `<DT>?<JJ|NN|NNS>*<NN|NNS>` (where `DT` stands for determiner; `JJ` for adjectives, and `NN|NNS` for nouns).

Title: Evaluating the performance of four snooping cache coherency protocols.
Authors: S. Eggers and R. Katz.
Publication Year: 1989.
Abstract: ... **coherency protocols** have been criticized for being unable to achieve **good bus performance** across **all cache configurations** ...

Transformed Token List: {coherency protocols, good bus performance, all cache configurations}

Figure 1: Example text from 1989, with noun phrases bolded.

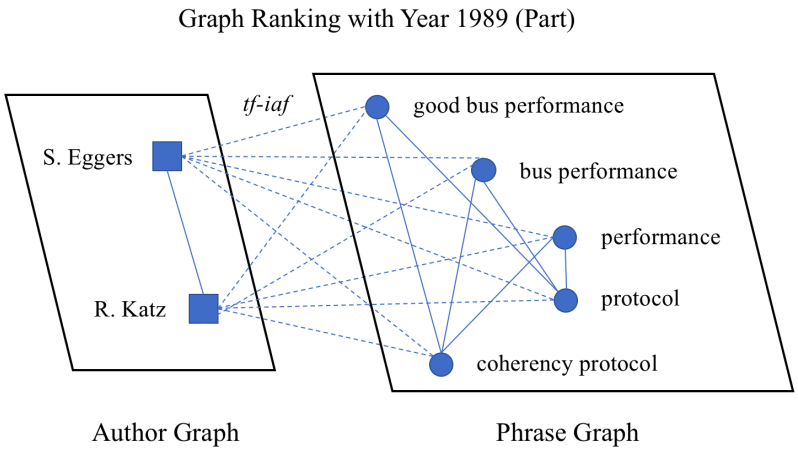

Figure 2: Portion of the multigraph derived from the text in Figure 1. Note that in the complete graph, nodes may connect with other unseen authors and phrases; we only show the part constructed by the sample document.

"phrase" graph: **coherency protocol** and **protocol**; similarly, **good bus performance**, **bus performance**, and **performance** from the next noun phrase. Subsequently, each node from {coherency protocol} will be linked with another node from {good bus performance}. Therefore, single nodes in our "phrase" graph can be entire noun phrases.

The "author" and "phrase" graphs are jointly connected based on the "author-writes-phrase" relationship, in which an author–phrase pair is denoted as { author a_i – phrase p_j}. We then propose *tf-iaf*, which is a variant of *tf-idf* (Salton et al., 1983) to calculate the weight of the edges:

$$\begin{aligned} tf\text{-}iaf_{a_i,p_j} &= tf_{a_i,p_j} \times iaf_{p_j} \\ &= \frac{Occ(a_i,p_j)}{\sum_{z=1}^{n} Occ(a_i,p_z)} \times \log \frac{|A|}{|A(p_j)|}, \end{aligned} \tag{1}$$

where $Occ(a_i, p_j)$, $|A(p_j)|$, and $|A|$ denote the number of occurrences of p_j by a_i, the number of authors who write p_j, and the total number of authors, respectively.

Returning to our running example, for the document group for year 1989, a portion of the graph is demonstrated in Figure 2.

Ranking and extraction of the key-authors and keyphrases can then be implemented through an iterative method, similar to the HITS algorithm. We use three matrices to store the "author", "phrase", and the *tf-iaf* graphs. More specifically,

· $\boldsymbol{N}_{m\times m}$ is a matrix for the set of authors $A = \{a_i | 1 \le i \le m\}$. Each element $n_{i,j}$ denotes how many papers auhtors a_i and a_j co-authored during the year. In the sample document presented above, if the authors "S. Eggers" and "R. Katz" have only collaborated on this paper in 1989, then we have $n_{Eggers,Katz} = 1$ for matrix $\boldsymbol{N}$.

· $\boldsymbol{W}_{n\times n}$ is a matrix for the set of phrases $P = \{p_j | 1 \le j \le n\}$. Each element $w_{i,j}$ denotes that how

often phrase p_i and p_j have co-occurred across all documents since that year. Note that for matrices $\boldsymbol{N}$ and $\boldsymbol{W}$, we have $\boldsymbol{N} = \boldsymbol{N}^T$ and $\boldsymbol{W} = \boldsymbol{W}^T$.

· $\boldsymbol{L}_{m \times n}$ is a matrix for the *tf-iaf* scores defined by Equation (1). Each element $l_{i,j}$ denotes the *tf-iaf* value, in other words, $l_{i,j} = tf\text{-}iaf_{a_i,p_j}$.

Additionally, we use a $m \times 1$ vector $\vec{i}$ and a $n \times 1$ vector $\vec{j}$ to represent the scores of the authors and phrases, respectively. Both vectors are initialized with 1 for each entry; subsequently, the score vectors $\vec{i}$ and $\vec{j}$ are updated using the following rule:

$$\begin{aligned} \vec{i} &= (\alpha \times \boldsymbol{N}\vec{i} + (1-\alpha) \times \boldsymbol{L}\vec{j}).normalized(), \\ \vec{j} &= (\alpha \times \boldsymbol{W}\vec{j} + (1-\alpha) \times \boldsymbol{L}^T\vec{i}).normalized(). \end{aligned}$$

where α is a tuning coefficient to determine the relative importance of authors and phrases during ranking.

Therefore, after the iterative ranking, noun phrases are assigned a score based on both the adjacency with other phrases and the author information for a particular year. Consolidating all scores for all phrases produces a $2D$ array, of which each row stands for the time series scores for a phrase.

3.2 Representativeness with Word Embeddings

A key point here is that the meaning of a phrase and its relation to other phrases evolves over time. For instance, during the development of the topic *Machine Learning*, the phrase "machine learning" was likely to be associated with popular supervised approaches such as "decision tree" and "SVM" around 2000, but after 2010 co-occurs more often with "deep neural networks". These changes indicate that the representativeness of a phrase to a topic also shifts over time. To model such dynamics, we use word embeddings to further refine the phrase scores from the previous step in our ETD pipeline.

Our model assigns a vector per phrase, per unit time step (year). We construct phrase vectors by averaging its individual word embeddings (originating from the 300-dimensional version of Google's Word2Vec model (Mikolov et al., 2013)). The phrase vector is calculated by taking the *tf-idf* weighted average of the vectors of individual words, where the *tf-idf* is calculated on the basis of the yearly-divided corpus. We note that we tried training dynamic word embeddings based on recalibrating the vectors as per many standard NLP application, but that technique did not lead to better performance.

As individual phrases are too fine-grained to be minted as trends, we apply a form of dimensionality reduction to find key phrases that can serve as a label for a trend. We group the vectors into clusters using the k-means algorithm, where each phrase is associated with a representativeness value calculated by taking the cosine similarity between its vector and its respective cluster center. The phrase score obtained from the graph ranking step is multiplied with the representativeness score. In this way, we assign phrases with the list of scores indicating their significance and representativeness over the years.

3.3 Predictions with RNN and Trending Phrase Ranking

For a particular phrase, given its scores over a certain period of time, we expect our model to predict the phrase score in the following time interval. In our model, we implement a basic recurrent neural networks (RNNs) to makethese predictions.

The model employs scores from the first k sequential years to predict $(k+1)^{th}$ score defined as:

$$x_{k+1} = f(x_1, x_2, x_3, ..., x_k).$$

For the array of scores from 1958 to 2015, this is a list of training samples where $\boldsymbol{x}$ is score vector that consists of k sequential years and $\boldsymbol{y}$ is the score in the $(k+1)^{th}$ year. The input of the RNN is $1 \times k$ representing the same feature (score of a certain year) spanned across k years, and the output is a single value representing the score in the $(k+1)^{th}$ year.

With the scores for each phrase predicted from year 1958 to 2015, it is then possible to investigate further into the phrases. As illustrated previously, it is expected that our model could make predictions of trending research fronts. Specifically, trends refer to research areas which are gradually being studied,

hence, there should be a strong foundation for them, meaning that there are certain level of existing work associated with them; yet they should not be the mostly-discussed issue at the same time. Based on these assumptions, we have defined a *trending score* for each phrase, which can be defined as follows:

$$trend = \beta \times (x_{k+1} - \max(x_1, \ldots, x_k)) + (1 - \beta) \times \max(x_1, \ldots, x_k),$$

where β is another tuning weight similar to that used in the graph ranking algorithm. Coupled with the previous steps' score computations, we can then determine the most influential and trending phrases in each year with the predictions from RNN. Our model selects the top ranked phrases based on their actual scores for each year, and produces another list based on predictions from RNN. It is expected that when comparing these two lists, phrases appearing near the top of the actual list also appear at the top of the prediction list.

4 Experiments

Evaluation of research trend detection is largely indicative, since the task is subjective, sources various, and replicability of prior work is difficult at best. Despite these problems, we propose a workflow that allows better future replicability, including the standardization of the datasets and the availability of the algorithms' prediction output[2].

In our experiments, we use the abstract of each paper and the published year as one document entry and the timestamp, respectively. After describing specific parameter settings, we describe our evaluation for the two different datasets in turn.

4.1 Parameter Settings

For the three sub-components of our model, we set the parameters as follows:

In the multi-graph ranking component, when iterating through the texts, phrases appearing within a window size of 3 are identified as adjacent tokens and are hence connected in the graph, when updating the final score vectors with Equation (1). We set α to 0.5.

When incorporating the Word2Vec model into phrase scores, we use the k-means algorithm to cluster the phrase vectors. We compare the performance of the model on various settings of $k = \{10, 15, 20, 25\}$. For the recurrent neural network predictions, we set a (context) block size of 3 for the input shape (*i.e.*, in the $2D$ array of scores, for each phrase, its time series scores incorporate the three prior values: $x_4 \leftarrow < x_1, x_2, x_3 >$).

The generated training samples are then fed into the RNN, which includes 2 hidden layers, where each layer has 4 neurons. We use mean squared error as the loss function (as the result is continuous), which naturally dictates the use of RMSProp as the optimizer. Following typical training regimes, we use a batch size of 10, and 80 epochs before termination. We also empirically tuned β to 0.8, emphasizing the positive and increasing trend in importance in defining trending topics and research fronts.

4.2 ACM Periodical Dataset

We use this dataset to evaluate the performance of multigraph ranking (MGR) component as we hypothesize that its mutual recursion will improve trend detection over the baseline TextRank source. This dataset includes around 9,740 XML files representing publications copyrighted by the Association of Computing Machinery (ACM), in the domain of computer science, published from 1958 to 2015. It is available from the ACM used by explicit permission. We use the abstract and metadata (specifically, publication time, authors, and keywords) for each paper. For space, in this presentation, we restrict the field of study to "software engineering"[3].

We evaluate our model with **Recall@n** and **nDCG@n**, and then compare it with the standard TextRank algorithm (Mihalcea and Tarau, 2004) as a baseline.

[2]Code is available at: `https://github.com/RichardeJiang/racode`.

[3]Filtered using the keyword "software engineering" and "software and its engineering" on the `keywords` and `concept description` component of each publication.

Recall @ n^{th} rank												
System	$k = 10$			$k = 15$			$k = 20$			$k = 25$		
	@10	@20	@30	@10	@20	@30	@10	@20	@30	@10	@20	@30
ETD-TR	11.3%	10.5%	8.5%	13.0%	12.9%	10.4%	7.6%	7.4%	6.5%	14.1%	13.2%	**10.5%**
ETD-M	**24.0%**	**21.9%**	**16.5%**	**25.9%**	**23.5%**	**21.1%**	**17.8%**	**15.6%**	**12.9%**	**16.7%**	**13.6%**	**10.5%**

Normalized Discounted Cumulative Gain (nDCG) @ n^{th} rank												
System	$k = 10$			$k = 15$			$k = 20$			$k = 25$		
	@10	@20	@30	@10	@20	@30	@10	@20	@30	@10	@20	@30
ETD-TR	0.105	0.127	0.138	0.107	0.136	0.143	0.078	0.090	0.095	**0.120**	**0.151**	0.161
ETD-M	**0.158**	**0.220**	**0.251**	**0.226**	**0.261**	**0.289**	**0.147**	**0.180**	**0.199**	0.107	0.136	**0.174**

Table 1: Performance comparison of Recall@n (top) and nDCG@n (bottom), varying the number of clusters $k = \{10, 15, 20, 25\}$ generated by k-means. Best system figures are bolded. "ETD-TR" and "ETD-M" stand for embedded trend detection with TextRank (Mihalcea and Tarau, 2004) (baseline) and our multigraph ranking, respectively.

Recall @ 20, $k = 20$					
Run	1	2	3	4	5
ETD-TR	7.4%	10.3%	8.6%	11.1%	9.2%
ETD-M	15.6%	14.0%	15.6%	13.2%	20.1%

Table 2: Comparison of Recall@20 with $k = 20$ clusters over different runs of our model.

Our embedded trend detection workflow consists of the three stages: keyphrase extraction, embedding representation, and final prediction with RNN. Our evaluation here tests the effectiveness when we change the first component to our proposed MGR. We replace the MGR with the standard TextRank algorithm and use the output as the baseline, and compare the resulting ranking model.

Recall@n measures how many predicted phrases are the actual top terms computed by the system. For each year, the first component (MGR and baseline TextRank) returns a list of the top terms. These are further enhanced through Word2Vec embeddings and fed to the RNN to generate top trending phrases. We first compute average recall based on this final output for all available years, and then compute normalized discounted cumulative gain (nDCG) (Järvelin et al., 2000) to further assess the quality of the element ranking. A system performs better if it returns true positive trending phrases towards the top of its rankings, as opposed to the bottom.

We test different settings of $k \in \{10, 15, 20, 25\}$ for the number of clusters when performing the k-means algorithm in the second Word2Vec stage for incorporation, as shown in Table 1.

From Table 1, we note that when extracting the keyphrases and make value predictions, our model consistently outperforms the baseline TextRank algorithm, in both the number of correctly predicted phrases (Recall) and their ordering (nDCG). Since the only difference between the baseline TextRank and our MGR model is how the phrases are ranked, we conclude that our multi-graph ranking algorithm yields more consistent scores in this dataset. We take this as evidence that in our MGR model, random interference is reduced when compared against the standard TextRank algorithm. This provides the empirical justification for our assumption when formulating the multi-graph ranking step – that important authors are more likely to cooperate with each other; and that words written by them are more likely to be trending in their respective fields. Additionally, the use of graphical techniques which find central components does reveal useful core nodes (concepts).

As the k-means algorithm does not guarantee to produce the same cluster result when we run the algorithm for several times, in order to show the consistency of performance, we have tested the system multiple times with the same set of parameters. Here, we show the example when $k = 20$ (number of clusters), and we compute the **Recall@20**, as illustrated in Table 2. We observe that results will be different when testing the system in different runs, which conform to the nature of k-means algorithm; however, our model (ETD-M) consistently outperforms the baseline ETD-TR with the same parameter settings.

Clusters	*Top 10 Keywords*
G_1	degree, growth, substrate, film, ga, gaas, gan, nh, si, surface
G_2	degree, contact, gan, al, ni, ga, ti, au, physics, american institute
G_3	gan, mg, layers, ga, physics, american institute, structure, defect, photoluminescence, strain

Table 3: Top 10 keywords detected by citation network methodology from (Shibata et al., 2008), for the year of 2000. Copied verbatim from the source.

Year	*Top 20 Keyphrases*
1999	molecular beam epitaxy, beam epitaxy, **substrate**, **surface**, epitaxy, deposition, results, quality, diffraction, metalorganic chemical vapor deposition, vapor deposition, **sapphire substrates**, conditions, cm, properties, measurements, chemical vapor deposition, electron microscopy, microscopy, spectra
2000	**growth**, **layers**, **substrate**, epitaxy, **surface**, diffraction, **substrates**, molecular beam epitaxy, electron microscopy, measurements, spectra, spectroscopy, microscopy, cm, properties, **structure**, quality, results, conditions, characteristics
2001	**growth**, **layers**, temperature, temperatures, **substrates**, emission, **layer**, epitaxy, measurements, spectroscopy, **surface**, molecular beam epitaxy, properties, diffraction, **films**, microscopy, beam epitaxy, cm, **photoluminescence**, **structure**

Table 4: Top keyphrases predicted by our ETD-M model. Phrases detected by both ETD-M and Shibata et al.'s (2008) model marked as bolded.

We also tested the sub-systems by eliminating the Word2Vec component from our pipeline of ETD: meaning that after we obtained the phrases together with their time series scores using graph ranking algorithms (either with TextRank or our proposed multigraph ranking, MGR), we feed the data into RNN and evaluate the performance. The resulting Recall@20 for TextRank and MGR are **14.5%** and **17.1%**, respectively. The values illustrate two inferences: firstly, even without the reinforcement of word embeddings, our proposed MGR still produces more consistent ranking scores. Secondly, if comparing the result with average Recall@20 from Table 1 (10.5%, 12.9%, 7.4%, 13.2% for TextRank, 21.9%, 23.5%, 15.6%, 13.6% for MGR), we observe that when the embedding component is added to our system, ETD-M outperforms ETD-TR when the number of k-means clusters is relatively small. By comparison, TextRank considers all available tokens, hence the adjacency relationship includes more information on the representativeness of each word. Therefore, in ETD-TR, we believe that word embeddings introduce noise to the original values. On the other hand, in MGR, the spatial information is diluted and Word2Vec helps to centralize the popularity of each phrase, especially when the number of clusters is small. This indicates that word embeddings are effective in our ETD-M model.

4.3 Science and Social Science Citation Index Datasets

In their paper, Shibata et al. (2008) provided the research fronts detected by their system for the domain of Gallium Nitride (GaN) – a prominent research field in applied physics and material science – for the year 2000 (Table 3). We wish to compare directly with their reported results, contrasting the top terms generated for phrases in the year range of 1999–2001, as shown in Table 4, hypothesizing that our ETD-M system can outperform their inferred keyphrase in terms of subjective keyphrase quality.

They used data gleaned from the Science Citation Index (SCI) and Social Science Citation Index (SSCI) original compiled by the Institute for Scientific Information, which are available from Web of Science databases. To compare with their results as best as possible, we have retrieved the papers using the same database query provided by their research: "GaN OR Gallium Nitride" to compare our results with theirs. We collected papers whose years of publication range from 1970 to 2004, to exactly match the parameters used in Shibata et al.'s work. The total number of the papers is around 15,200.

We note that about half of the keyphrases obtained by Shibata et al.'s work are also predicted suc-

cessfully by the ETD-M model. Examining the two tables carefully, we believe that phrases from our model provide better quality insight: element names such as "ga", "ni", and "al" recognized by Shibata et al.'s work unfortunately provide no detail on the study; furthermore, some terms recognized by them like "american institute" are also irrelevant to the research topic.

Since Shibata et al.'s work filters out non-core papers first, potential keywords can only originate from core papers. According to their results, out of 4,000 publications collected in the year 2000, around 30 are selected based on the roles of each paper within its cluster. Although we cannot replicate their work exactly, we surmise that these central papers may manifest such terms with sufficiently high frequency to be extracted, resulting in the large involvement of element names and non-relevant terms in the output. Due to the mutual recursion element of ETD-M, we can propagate the influence of keyphrases beyond such a core, while assigning an appropriate level of credit. Such results allow ETD-M to convey and rank longer, meaningful phrases such as "molecular beam epitaxy" and "metalorganic chemical vapor deposition" within that domain.

5 Future Work

The three-stage pipeline in our ETD-M model is designed to provide insights on the trending research keywords and has been proven to work better than the standard TextRank and bibliometric approaches. Additionally, the model can be improved from various perspectives.

In our multigraph ranking (MGR) step, the score vectors for authors and phrases are normalized after each iteration of updates to limit the values within the range of $[0, 1]$; the matrices representing author and phrase mutual relations ($\boldsymbol{N}, \boldsymbol{W}, \boldsymbol{L}$) are non-smoothed. We expect by normalizing matrices $\boldsymbol{N}, \boldsymbol{W}, \boldsymbol{L}$ to $\hat{\boldsymbol{N}}, \hat{\boldsymbol{W}}, \hat{\boldsymbol{L}}$ in the first place, and then using the normalized matrices in the ranking, the output score vector will therefore indicate the relative importance and the value will be smoothed. The update rules hence become:

$$\begin{aligned} \vec{i} &= \alpha \times \hat{\boldsymbol{N}}\vec{i} + (1-\alpha) \times \hat{\boldsymbol{L}}\vec{j}, \\ \vec{j} &= \alpha \times \hat{\boldsymbol{W}}\vec{j} + (1-\alpha) \times \hat{\boldsymbol{L}}^T\vec{i}. \end{aligned}$$

Our model uses Google's Word2Vec to enhance the representativeness of each phrase in every year by taking the cosine similarities between the vector representation of a phrase and its respective cluster center. Given a specific collection of documents, the embedding for its vocabulary can be rather different from the fixed, pretrained Word2Vec, so in the future we may consider further training the existing Word2Vec with our yearly-divided data. In addition, the representativeness of phrases can also be extended by incorporating topic models, so as to assign a topic and a value to each phrase would admit the relative importance of the phrase, while providing an intuitive approach in clustering.

The evaluation of our model is largely subjective: whether the output keyphrase list provides insights with better quality depends on human interpretation. To better understand the efficacy of our model, it can be applied to other disciplines such as using the PubMed data, and then we employ domain experts to manually evaluate the keyphrase list.

One common issue with a pipeline model is that error may propagate through different stages and accumulate. Therefore, the overall structure of the model can be redesigned such that the graphs are already time-aware during ranking. Wang et al. (2013) have provided an article ranking model which takes publication time into account. When applying to our model, we could introduce a "time" graph to the existing MGR model, where the heterogeneous graph incorporates publication times.

6 Conclusion

We have presented our ETD-M model for detecting research fronts and trends from scientific publications. Unlike prior work that emphasize evidence from citation links and topological measures, our model mines salient phrases from the abstract of the publications themselves. Our three-stage ETD-M pipeline features a multigraph keyphrase ranking (MGR) algorithm, followed by k-means clustering and keyphrase representation utilizing word embeddings, with a final stage of detecting the topics using an RNN for the final, temporally-sensitive inference of trending topics.

A key contribution from our work lies in the first stage of evidence gathering, in the form of our keyphrase extraction. We structure this task as a mutual recurrence between important authors and their keyphrases. We compare our MGR against the TextRank algorithm within the ETD framework, which clearly showed the advantages of utilizing author information in identifying salient keyphrases. Additionally, we feel that our MGR model's inferred research fronts align well with historical fact.

Our ETD-M model can be extended further for a more comprehensive structure. Applying normalized author and phrase matrices to the update rule makes the scores more reasonable; Mikolov's Word2Vec can be adjusted to fit in our own dataset for each year. Additionally, we can introduce human evaluation by feeding in data from various fields of study, to further evaluate the efficacy of our model while discovering trend insights to these other scientific research areas.

Acknowledgments

This research is supported in part by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative. In addition, our experiments are carried out based on the ACM Digital Library dataset (with the metadata state dated 11 September 2015) furnished in collaboration with the Association for Computing Machinery.

References

David M. Blei, Andrew Ng, Michael Jordan, and John Lafferty. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022.

David M. Blei and John Lafferty. 2006. Dynamic Topic Models. In *Proc. of the 23rd Conference on Machine Learning (ICML'06)*, pages 113–120.

Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and C. Lee Giles. 2009. Detecting Topic Evolution in Scientific Literature: How Can Citations Help? In *Proc. of the 18th International Conference on Information and Knowledge Management (CIKM'09)*, pages 957–966.

Aaron Elkiss, Siwei Shen, Anthony Fader, Gunes Erkan, David States, and Dragomir Radev. 2008. Blind Men and Elephants: What Do Citation Summaries Tell Us About a Research Article? *Journal of the American Society for Information Science and Technology (JASIST)*, 59(1):51-62, 2008.

Katsuhide Fujita, Yuya Kajikawa, Junichiro Mori, and Ichiro Sakata. 2012. Detecting Research Fronts Using Different Types of Weighted Citation Networks. *Journal of Enginnering and Technology Management (JET-M)*, 32:129–146.

Takao Furukawa, Kaori Mori, Kazuma Arino, Kazuhiro Hayashi, and Nobuyuki Shirakawa. 2014. Identifying the Evolutionary Process of Emerging Technologies: A Chronological Network Analysis of World Wide Web Conference Sessions. *Technological Forecasting and Social Changes*, 91:280–294.

David Hall, Daniel Jurafsky, and Christopher Manning. 2008. Studying the History of Ideas Using Topic Models. In *Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 363–371.

John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. 2004. Tracking Evolving Community in Large Linked Networks. In *Proc. of the National Academy of Sciences (PNAS)*, 101(suppl.1):5249–5253.

Kalervo. Järvelin and Jaana. Kekäläinen. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pages 41–48.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1746–1751.

Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM)*, 46(5):604-632, 1999.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 404–411.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of the Workshop at the 1st International Conference on Learning Representations (ICLR 2013)*.

Fabian Mörchen, Mathäus Dejori, Dmitriy Fradkin, Julien Etienne, Bernd Wachmann, and Markus Bundschus. 2008. Anticipating Annotations and Emerging Trends in Biomedical Literature. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, pages 954–962.

Mark E. J. Newman. 2004. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69(6):066133.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. In *Technical Report SIDL-WP-1999-0120*, Stanford Digital Library Technological Project.

Derek J. De Solla Price. 1965. Networks of Scientific Papers. *Science*, 149(3683):510–515.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The Author-Topic Model for Authors and Documents. In *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, pages 487–494.

Gerald Salton and Michael J. McGill. 1983. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

Naoki Shibata, Yuya Kajikawa, Yoshiyuki Takeda, and Katsumori Matsushima. 2008. Detecting Emerging Research Fronts Based on Topological Measures in Citation Networks of Scientific Publications. *Technovation*, pages 758–775.

Qi Wang. 2017. A Bibliometric Model for Identifying Emerging Research Topics. *Journal of the Association for Information Science and Technology*, 69(2):290–304.

Yujing Wang, Yunhai Tong, and Ming Zeng. 2013. Ranking Scientific Articles by Exploiting Citations, Authors, Journals, and Time Information. In *Proc. of the 27th AAAI Conference on Artificial Intelligence (AAAI-13)*, pages 933–939.

Ian Witten, Gordon Paynter, Eibe Frank, Carl Gutwin, and Craig Nevill-Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proc. of the 4th ACM Conference on Digital Libraries (DL '99)*, pages 254–255.

Ying Yan, Qingping Tan, Qinzheng Xie, Ping Zeng, and Panpan Li. 2017. A Graph-based Approach of Automatic Keyphrase Extraction. *Procedia Computer Science, 107 (C)*:248–255.

Yuxiang Zhang, Yaocheng Chang, Xiaoqing Liu, Sujatha Das Gollapalli, Xiaoli Li, and Chunjing Xiao. 2017. MIKE: Keyphrase Extraction by Integrating Multidimensional Information In *Proc. of the 27th International Conference on Information and Knowledge Management (CIKM'17)*, pages 1349–1358.