Embedding-based Detection and Extraction of Research Topics from Academic Documents Using Deep Clustering

Sahand Vahidnia[†], Alireza Abbasi, Hussein A. Abbass

University of New South Wales, Canberra, 2612, ACT, Australia

Abstract

Purpose: Detection of research fields or topics and understanding the dynamics help the scientific community in their decisions regarding the establishment of scientific fields. This also helps in having a better collaboration with governments and businesses. This study aims to investigate the development of research fields over time, translating it into a topic detection problem.

Design/methodology/approach: To achieve the objectives, we propose a modified deep clustering method to detect research trends from the abstracts and titles of academic documents. Document embedding approaches are utilized to transform documents into vector-based representations. The proposed method is evaluated by comparing it with a combination of different embedding and clustering approaches and the classical topic modeling algorithms (i.e. LDA) against a benchmark dataset. A case study is also conducted exploring the evolution of Artificial Intelligence (AI) detecting the research topics or sub-fields in related AI publications.

Findings: Evaluating the performance of the proposed method using clustering performance indicators reflects that our proposed method outperforms similar approaches against the benchmark dataset. Using the proposed method, we also show how the topics have evolved in the period of the recent 30 years, taking advantage of a keyword extraction method for cluster tagging and labeling, demonstrating the context of the topics.

Research limitations: We noticed that it is not possible to generalize one solution for all downstream tasks. Hence, it is required to fine-tune or optimize the solutions for each task and even datasets. In addition, interpretation of cluster labels can be subjective and vary based on the readers' opinions. It is also very difficult to evaluate the labeling techniques, rendering the explanation of the clusters further limited.

Practical implications: As demonstrated in the case study, we show that in a real-world example, how the proposed method would enable the researchers and reviewers of the academic research to detect, summarize, analyze, and visualize research topics from decades

Citation: Vahidnia, S., Abbasi, A., & Abbass, H.A. (2021). Embedding-based detection and extraction of research topics from academic documents using deep clustering. *Journal of Data and Information Science*, 6(3), 99–122. <https://doi.org/10.2478/jdis-2021-0024>

Received: Nov. 30, 2020

Revised: Feb. 23, 2021;

Apr. 26, 2021

Accepted: Apr. 26, 2021

[†] Corresponding author: Sahand Vahidnia (E-mail: s.vahidnia@unsw.edu.au, vahidnia.sahand@gmail.com).

Research Paper

of academic documents. This helps the scientific community and all related organizations in fast and effective analysis of the fields, by establishing and explaining the topics.

Originality/value: In this study, we introduce a modified and tuned deep embedding clustering coupled with Doc2Vec representations for topic extraction. We also use a concept extraction method as a labeling approach in this study. The effectiveness of the method has been evaluated in a case study of AI publications, where we analyze the AI topics during the past three decades.

Keywords Dynamics of science; Science mapping; Document clustering; Artificial intelligence; Deep learning

1 Introduction

Detection of research fields or topics in science and their dynamics overtime is an active field of research. This field falls under the field of Science of Science (SciSci), aiming to understand, quantify and predict scientific research dynamics and the drivers of that dynamics in different forms such as the birth and death of scientific fields and/or their sub-fields (Zeng et al., 2017) that can be identified by tracking the changes of research trends. This helps governments, businesses, and scientists in their decisions regarding establishing the fields of science and investment in the fields, contributing to the research budgets. The automation of this process known as topic detection, gives the scientists a faster way to discover, summarize, and represent the research topics based on a large corpus of documents, independent of the subjective opinions, as opposed to reviewing thousands of documents. It also paves the path to predictive models.

There have been varying methods proposed and explored in the literature to analyze and understand the dynamics of science considering the change of scientific fields and their sub-fields. Topic modeling techniques such as Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) and Latent Semantic Analysis (LSA) (Deerwester et al., 1990) are amongst the most popular methods in the field that are used to understand relationships among the contextual information and text documents (Jelodar et al., 2019). In addition, network-based techniques such as word co-occurrence, and citation networks are one of the most explored methods in the literature for revealing relationships in data (Hou, Yang, & Chen, 2018; Kuhn, Perc, & Helbing, 2014; Liu et al., 2019). However, after recent developments in machine learning and natural language processing (NLP), new methods in text mining such as word and document embeddings have facilitated analyzing the metadata (Joulin et al., Mikolov et al., 2013) or contents of publications to understand the dynamics of the fields (Zhang et al., 2017).

An approach to detect and extract topics is to cluster the documents. This requires us to represent the data in vector space. This representation may be extracted using various methods (e.g. Word2Vec (W2V) (Mikolov et al., 2013)), each having advantages and disadvantages. These representations can also be conveyed as features. The clustering stage itself does also require investigation to find the best algorithm and model, fit for the embedding methods. It has been demonstrated in numerous studies that NLP and word and document embeddings may impact the outcome and yield improvements in the results in a variety of downstream tasks (Curiskis et al., 2020; Radu et al., 2020; Weber et al., 2020; Zhang et al., 2018), specifically compared to the classical and topic modeling methods. Hence, it is important to explore thoroughly the models and methods of utilizing text embeddings for topic detection and extraction. However, having multiple varieties of embedding methods and different clustering approaches, and the challenges in understanding the context of the extracted topics make the utilization of these methods less straightforward. As has been shown in this study, some methods may be fit only for some specific tasks. Therefore, researchers rely on more straightforward and established methods and tools such as topic modeling using LDA. Hence, extraction of topics from the abstract of scientific documents requires an established path. Existing embedding and clustering-based methodologies in the literature require tuning to extract topics from academic document metadata such as their title and abstract texts. This can be regarded as the matter of understanding the features useful to our task, and only exploiting them. In addition, an established framework and combination of models and methods are required in the literature for future studies of topic detection from document abstracts, to pave the way and serve as an established baseline. Hence we devise a plan to meet the main objective of this study, which is detection of the research topics within a field, enabling further analysis including temporal tracking. This study will analyze the state of the art and the classical methods in clustering and embedding to establish a research path for improving the status quo and a reference for future studies.

The remainder of the paper is organized as follows: First, we review the literature surrounding the approaches and stages of the objective (i.e. Feature extraction, categorization, and labeling). Then we discuss the methodology and the evaluation metrics. The results are discussed next and the methods are compared to the proposed method. Finally, a case study has been conducted to analyze the activity of sub-fields or research areas in three AI journals, throughout the study period.

2 Review of topic detection processes and approaches

A way of detection of the topics within fields of science is to extract topics from textual data, using either topic modeling approaches (Jelodar et al., 2019; Xu, Zhai, et al., 2019; Xu et al., 2019; Zhou et al., 2019) or clustering approaches (Curiskis et al., 2020; Kim et al., 2020; Radu et al., 2020; Zhang, Chen, et al., 2017). The overall process, considering the clustering approach in this study can be simplified and put into stages including Document Embedding, Document categorization and clustering, and Labeling of the categories, illustrated in Figure 1. The topic modeling approach on the other hand does replace the third and the fourth stages with topic modeling. These stages are especially important in this study to aid in quantitatively comparing and analyzing the methods involved in each step for the task of this study, which is clustering. This parameterizes the task stages, including the embedding and categorization stages for a comprehensive comparison and model selection. In the following, based on our perspective of the field, popularity, and the objectives, we review and explain the literature surrounding the topics of “Embedding”, “Categorization”, and “Labeling”. These three topics are the necessary stages of the objective of this study. Hence, other aspects of the field including classification, recommendation, and author activity analysis are skipped or less focused.

Figure 1. The overall and simplified process and objectives.

2.1 Text feature representation: Embedding

Performing operations on text data requires us to represent the textual data in meaningful numerical form, and generally fixed size (Le & Mikolov, 2014), for algorithms to understand. This process, which enables mathematical operations on text, is referred to as vectorization or embedding.

Classic embedding methods:

Simple representations such as Bag-of-Words (BoW) have been proposed very early in the field (Harris, 1954), which serve as the most basic approach for embedding texts to vectors. These representations can bring the sentences to a vector space, but have important limitations. They have very high dimensionality and are sparse. In BoW representation, the dimension of the vectors are the same as the number of unique words in the corpus, and each sentence/paragraph can be

represented by the number of each word in the vocabulary (or only the number 1 if we are not interested in the number of hits), which will result in many zeros. Aside from the shortcomings, these embeddings can be used as the foundation of the more complex embedding methods, i.e. neural network models like W2V (Mikolov et al., 2013) and Siamese (Kenter, Borisov, & De Rijke, 2016). Some other methods incorporate simple statistical vectors like TF-IDF (term frequency—inverse document frequency) (Jones, 1972). Yet, they lack the rich and necessary information regarding the data and are very inefficient (Vahidnia, Abbasi, & Abbass, 2020). However, in addition to their usage in text representation (Zhang, Li et al., 2017; Zhang, Chen, et al., 2017), it is also very common to use TF-IDF for keyword extraction, as it has been incorporated in literature for similar purposes (Awan & Beg, 2020; Peganova, Rebrova, & Nedumov, 2019; Radu et al., 2020).

Deep embedding methods:

There are different approaches to have embeddings for academic documents, as embeddings can be either for words, documents, or network structures. The embedding type fit for the objective of this study is representing textual data in vector space. Embedding approaches include ways to bring high-dimensional sparse vectors to low-dimensional dense space. These vectors then can be used as features for downstream tasks, including both supervised and unsupervised learning tasks. Hence, the embedding stage can be regarded as the feature extraction stage. Word and document embedding as a methodology has come a long way and it goes under the natural language processing (NLP) field. Hence, the literature surrounding it can be divided into two categories: 1) pure NLP such as W2V (Mikolov et al., 2013) and regardless of the downstream tasks (i.e. classification, keyword extraction, or clustering); and 2) specialized and tuned for SciSci tasks such as SciBERT (Beltagy, Lo, & Cohan, 2019).

Models like W2V (Mikolov et al., 2013) were introduced to enrich the vectors with increasing the density and lowering dimensionality, by providing more accurate word representations. W2V has been used in numerous studies (Cagliero & La Quatra, 2020; Weber et al., 2020; Zhang et al., 2018) for a variety of downstream tasks, including text and word classification and clustering. W2V was complemented by the introduction Doc2Vec (Le & Mikolov, 2014), a method to embed the documents in a very similar fashion to W2V, by adding paragraph id to the neural network input. After the introduction of W2V, other methods also emerged, including FastText (Joulin et al., 2016), which enriches the embeddings with sub-word information. This enables FastText to be able to embed the unseen words based on the subword information.

The most recent and advanced methods for embedding are developed thanks to BERT (Devlin et al., 2018) and the introduction of transformers. Their biggest advantage is their reusability and good response to fine-tuning. Another major embedding method introduced parallel to BERT is ELMo (Peters et al., 2018) that uses bidirectional Long Short Term Memory or LSTM to learn word representations as a function of a sentence. ELMo is known to provide very robust word representations, aware of its surrounding context. However, using these methods are not as straightforward and are generally more resource intensive. In addition, BERT vectors have considerable larger dimension as opposed to the previous method, which adds to the difficulty of implementation.

2.2 Categorization

Identification of science entities such as the fields, sub-fields, topics, or concepts is described as a categorization problem in this study. In literature, the categorization problem is generally addressed in three ways: Topic modeling, Concept Extraction, and Clustering. All three approaches can provide ways to explore the fields of science by categorizing the textual data, and provide insights into the context. The clustering approaches and some concept extraction approaches may require the textual data to be embedded into vectors.

2.2.1 Topic modeling

One of the most common directions is to solve it as a topic modeling problem, which involves methods to discover the patterns of words used within documents, and to understand relationships among data and text documents (Jelodar et al., 2019). These methods provide ways to categorize, annotate, and understand large volumes of user-generated data (Curiskis et al., 2020). Topic modeling methods, specifically the methods derived from LDA, have been used in many recent and past studies for scientific document analysis, including (Xu et al., 2019; Xu, Zhai, et al., 2019; Zhou et al., 2019; Xu et al., 2020). An advantage of these approaches is that they do not require the text to be embedded into vectors, as they pursue to find statistical relationships among documents, topics and the terms.

2.2.2 Clustering

The other popular approach is to turn the problem into a clustering or an unsupervised learning problem. It can be divided into document clustering, term clustering, and concept or keyword clustering problems. Clustering is a task of automatically organizing entities such as documents into groups, such that entities within the same group are similar when as opposed to the entities in other groups,

with machine learning techniques. Similar to embedding, most methodologies are not originally intended for SciSci. Hence, the literature comprises the studies unrelated to SciSci or close to social networks (SN) as the downstream tasks. The literature around clustering varies based on the aspects of the problem that have been studied. Some studies focus on the clustering algorithms, while others focus on the overall approaches.

2.3 Combined document embedding and clustering

Among the closely related studies, Zhang et al. (Zhang, Chen, et al., 2017) in a study propose a document clustering approach in a sequential period, originally introduced in (Zhang, Zhang et al., 2017). The study explores the topics in academic publication and their births, deaths, and overall evolution. The author then extends the study in another study (Zhang et al., 2018) to take advantage of deep embedding methods as feature extraction, e.g. W2V.

Xie, Girshick, and Farhadi (2016) proposes a method to improve the unsupervised clustering performance, named Deep Embedded Clustering (DEC). DEC is a deep neural networks based method designed for clustering, leveraging dimensionality reduction power of the neural networks. Kim et al. (2020) in a patent document clustering study, utilizes DEC. They use Doc2Vec (Le & Mikolov, 2014) to acquire the patent document vectors, then feed the document vectors to an autoencoder as the pre-training stage. The output of the final layer of the encoder, also known as the code, is then fed into another neural network to perform the clustering. The initialization stage for the cluster centroids is done using k-means algorithms in this method. The method could outperform other methods including Doc2Vec with K-means and Doc2Vec with Gaussian Mixture Models, as clustering methods.

In a comparison study (Curiskis et al., 2020), different clustering approaches for clustering documents on online social networks have been compared and quantitatively evaluated. The evaluation and comparison include the embedding approaches coupled with well-known clustering algorithms, and also compared to topic modeling approaches. The study concludes that for document clustering, document embedding is the optimal solution. It also has been shown that K-means outperforms other conventional methods like hierarchical clustering and non-negative matrix factorization.

In another document of embedding and clustering comparison study (Radu et al., 2020), authors compare K-means, Spherical K-means, LDA and Density-based spatial clustering of applications with noise (DBSCAN) (Ester et al., 1996) as clustering algorithms by Doc2Vec and TF-IDF features. The study concludes that Doc2Vec improves cluster compactness.

2.4 Labeling of the clusters

A simple, yet a very smart method for labeling the clusters is to get TF-IDF based keywords of clusters and weight them based on similarity to the centroid of each cluster (Curiskis et al., 2020). Other methods are more advanced, taking advantage of ranking algorithms (e.g. TOP-Rank (Awan & Beg, 2020)). ComboBasic (Astrakhantsev, 2015). HCBASIC (Peganova et al., 2019) is an adaptation of ComboBasic, also taking the hierarchical structure of clusters into account. Rapid Automatic Keyword Extraction (RAKE) (Rose et al., 2010) is another key-phrase extraction method, which has been utilized successfully in numerous studies for similar purposes, such as a study by Krenn and Zeilinger (Krenn & Zeilinger, 2020), for “concept extraction”.

3 Methodology

In order to detect topics, and to observe the progress of them in time, we devise a framework, comprising the following steps: *i*) Data acquisition, *ii*) data pre-processing, including preparation of the data for training and embedding, *iii*) Document Embedding, by training the data utilizing different methods and parameters; *iv*) Categorization and clustering, by applying a selection of popular and relevant clustering techniques to document vectors. As a baseline method, LDA topic modeling has also been used in this study to compare to the proposed method, and generally compare LDA to embedding and clustering approaches. Later, the clustering results are evaluated against the ground-truth to select the best model. After model selection and as the final stage, *v*) we use a labeling method to assign labels to the context of the clusters.

3.1 Data pre-processing

This stage is about preparing the textual data for training and analyzing, which includes stop-word removal, removal of frequent abbreviations (like “e.g.”, “i.e.”, and “etc.”), removal of technical abbreviations, unifying English words to American English, numeric value removal, punctuation cleanup, and lemmatization by WordNet (Miller, 1995) using an NLP library.

3.2 Topic modeling

Topic models are statistical methods to uncover the patterns and clusters of topics in the document. LDA and LSA are the most known and baseline topic models. LDA has proven very effective across many studies. Hence, an implementation of LDA is used in this study as a baseline for comparing to clustering methods. LDA is a generative probabilistic model, assuming a Dirichlet prior over the latent topics

(Blei et al., 2003). LDA analyzes the relationships in a set of documents and their terms and produces a set of concepts. LDA can perform very well in clustering documents or generating topics, yet many topics may not be easily labeled or distinguished.

3.3 Embedding

Embedding the documents allows us to represent the documents in a low dimensional, dense vector space. Hence, this stage can also be denoted as the feature representation stage. Document embedding is necessary for enabling us to work with the documents and performing mathematical operations on them. Document embedding methods are not much different from word embedding methods and they also take advantage of word embedding at heart, with some modifications. Document vectors can be acquired in multiple ways, including aggregation of word embeddings, document embeddings (Doc2Vec), etc. In this section we describe the embedding approaches implemented for evaluation and comparison.

3.3.1 Aggregation of word embeddings: FastText

These are a set of methods utilizing word embeddings to acquire document vectors. Basically, most methods even the transformers use word embeddings at heart, but in this section, we will only rely on FastText embeddings, as it has proven very accurate in word embeddings due to its utilization of sub-word information. Using the mean of the word embeddings in a document is an intuitive baseline for acquiring document vectors (Vahidnia et al., 2020). Hence, as a baseline method, it is essential to incorporate this method and observe the competition. Alternatively, one may use smoothed embeddings to enhance the document vectors. Using a method based on SIF (Arora, Liang, & Ma, 2017) to give higher weights to the words with a higher importance in a document can potentially increase the document vector accuracy, by better representing the context. The incorporated adaptation of this method is similar to (Vahidnia et al., 2020), as demonstrated at Equation (1). Here, $wv(t)$ is calculated for each word for all documents, and then is divided by the number of words in the corresponding document. $v(t)$ represents the word vectors, $p(t)$ is the probability of seeing that word, and α is a constant $1e-3$. The word window size in the training instances for FastText is fixed at 5 and the vector dimensions for the FastText word embeddings are 100.

$$w(t) = \frac{\alpha}{\alpha + p(t)} \quad (1)$$

$$wv(t) = w(t) * v(t) \quad (2)$$

3.3.2 Document embedding

In this study, Doc2Vec (Le & Mikolov, 2014) is used as the main document embedding approach. Doc2Vec is a well-known and successful document embedding method, which is structurally very similar to W2V. However, it also takes the paragraph as a whole into account by designating an ID to each paragraph when training the model. This enables the Doc2Vec models to be aware of the context of the whole paragraph or document, instead of the training window. Similar to the FastText models, 100-dimensional embeddings are generated using a word window of 5 in Doc2Vec models. In the Doc2Vec models, the distributed memory model or PV-DM is used, as opposed to the distributed bag of words (DBOW) model. In the PV-DM model, the task is to predict a center word based on the surrounding context words and paragraph ID as depicted in Figure 2, where the DBOW model tries to predict paragraph words based on the paragraph ID, ignoring the context words (Le & Mikolov, 2014). Bidirectional Encoder Representations from Transformers or BERT (Devlin et al., 2018) is another document embedding method employed in this study. Compared to the other methods, BERT is more resource-intensive, and fine-tuning is not an easy task. However, pre-trained BERT models are very popular and are known to achieve the state of the art performance in numerous downstream tasks and datasets, including relation extraction and sentence classification tasks (Beltagy et al., 2019). Hence, this motivates us to use the following two pre-trained models in this study: i) English uncased BERT (A12, H768, L12) (Devlin et al., 2018), and ii) SciBERT (Beltagy et al., 2019) model, pre-trained on scientific corpora.

Figure 2. PV-DM model diagram.

3.4 Clustering

Document clustering can be carried out in various ways, using off the shelf or modified methods including the popular K-means (and its derivatives), hierarchical agglomerative clustering with Ward's method (Ward Jr, 1963), DBSCAN, DEC (Xie et al., 2016). In this study, the clustering techniques are put into test, tuned, and evaluated to find the optimal solution for clustering and categorizing the documents to represent the fields and sub-fields. To atone the randomness, K-means clustering has been repeated with multiple seeds in order to achieve optimal performance, incorporating both random and K-means++ initialization methods.

Similarly, the modified DEC is trained on multiple seeds, as DEC may have variation in output as a result of randomness in initialization and optimization steps. In this study, we use a modified DEC approach for clustering. DEC works in two steps: (1) parameter initialization, and (2) parameter optimization (Xie et al., 2016). Initially, it maps the data space X non-linearly to a latent feature space of Z , using the θ parameter. To achieve this, a pre-training of the data on Deep Autoencoders is conducted to encode the input vectors to output the features for clustering. The output of the encoder, also known as the code, is the latent feature space Z . Then, iteratively optimizes the parameters by minimizing the Kullback–Leibler (KL) divergence loss (Equation (3)) between the probability of assigning a sample i to a cluster j (soft assignment) Q , and the auxiliary distribution function P (Equation (4)), while cluster centers are initialized using K-means, and the parameters are initialized in previous step (Xie et al., 2016). The modified DEC model in this study has four sequential encoder layers with 200, 500, and 10 layers (Figure 3), with ReLU as activation. The Autoencoder is optimized on mean square error (MSE).

$$L = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3)$$

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \quad (4)$$

$$f_j = \sum_i q_{ij} \quad (5)$$

Hierarchical agglomerative clustering is a bottom-up clustering approach, invariant to the randomness. It has been observed previously (Vahidnia et al., 2020) that Ward's method for node distance measure is the best approach fit for this task in hierarchical agglomerative clustering. As opposed to the previous approaches (i.e. DEC, K-means, and Hierarchical), DBSCAN, a density based clustering algorithm, does not require the number of clusters (k) to initialize the clustering. Hence, there

Figure 3. DEC Autoencoder diagram. K is the number of clusters.

are two main variables in DBSCAN: minimum points and eps. The minimum points number defines the minimum number of points to be considered a cluster. The value of eps defines the distance threshold of points from a cluster, to be included in the cluster. In this study we use a range of eps from 0.01 to 1.0, to cover all the possible distances and record the best results.

4 Evaluation

4.1 Dataset

The study requires document data, with topics or categories as labels. There are various data available to use for this purpose. However, most datasets are less relevant to scientific documents or have unfit labels for evaluation purposes. Thus, choosing the appropriate dataset is a crucial step to correctly validate the methodology. One of the suitable datasets for the task of this study is KIPRIS dataset[Ⓢ]; a dataset of patent documents having 5 categories, also compiled and used in (Kim et al., 2020). The dataset comprises 19,710 patents. To enrich the training corpus with more textual data, the concatenated titles and abstracts of 14,000 the most used articles are downloaded from the Web of Science (WoS) database. We used the Computer Science category as the filter for the WoS search, and sorted

the data in descending order based on their usage number in WoS index. This provided a balanced number of publications with low and high citation counts, covering studies since 1956 to 2021, with most publications in 2010 to 2020 period (Figure 4). This will show how an addition of less relevant dataset with more term diversity will impact the result.

Figure 4. Additional WoS data figure.

4.2 Evaluation metrics

Performance evaluation is done using common metrics for unsupervised learning tasks, including Normalised Mutual Information (NMI), Adjusted Mutual Information (AMI), and also Adjusted Rand Index (ARI). NMI and AMI are mutual information-based metrics to measure the assignment agreement, ignoring the permutation. The upper bound 1 indicates a perfect match, and lower values are not as good. ARI measures the similarity of two assignments regardless of the permutations and cluster structures. ARI scores have a bounded range, where 1 indicates a perfect match and 0 means complete disagreement. As ARI and AMI are adjusted for chance, hence are good metrics for this comparison. Hence, in this study, we chose to use ARI for evaluation and model selection, as it can provide a consensus index for model selection (Steinley, 2004), followed by AMI and NMI. ARI can be defined as follows:

$$RI = \frac{a + b}{C_2^n} \quad (6)$$

$$ARI = \frac{RI - e[RI]}{\max(RI) - e[RI]} \quad (7)$$

Where a is the number of element pairs that are in both a ground truth class set and a cluster set. b is the number of elements in different ground-truth class sets and cluster sets. C_2^n denotes the number of possible pairs in the data. $e[RI]$ denotes the expected RI of random labeling.

The silhouette coefficient is the difference of the distance mean between a sample and all other points in the next nearest cluster, and the mean distance between a

Research Paper

sample and all other points in the same cluster, over the maximum of them (Rousseeuw, 1987). Silhouette coefficient can be very informative in certain scenarios, especially when the ground truth data is unavailable. However, in this study we have access to the ground-truth data. Hence, the coefficient will not be used as an evaluation metric in this study.

4.3 Results

As discussed in the previous section, to establish a methodology for topic extraction, a series of embedding and clustering methods have been tested and compared against each other and LDA as a baseline method. The result of the comparison has been recorded in Table 1.

Table 1. Comparison of clustering techniques with various embeddings on two different training corpora.

Training Dataset =>		KIPRIS			WoS+KIPRIS		
Evaluation metrics=>		NMI	AMI	ARI	NMI	AMI	ARI
Embedding	Clustering						
6*FastText (mean)	K-means (rand)	0.379	0.379	0.312	0.387	0.387	0.327
	K-means (++)	0.379	0.379	0.322	0.387	0.387	0.327
	Hierarchy Aggl.	0.391	0.391	0.289	0.363	0.363	0.306
	DBSCAN	0.006	0.005	0.000	0.005	0.005	0.000
	DEC	0.511	0.511	0.504	0.459	0.459	0.400
	DEC (scaled)	0.329	0.329	0.268	0.284	0.283	0.239
6*FastText (w. mean)	K-means (rand)	0.243	0.243	0.186	0.239	0.239	0.184
	K-means (++)	0.243	0.243	0.186	0.239	0.239	0.184
	Hierarchy Aggl.	0.260	0.260	0.140	0.234	0.234	0.176
	DBSCAN	0.037	0.035	0.001	0.011	0.010	0.000
	DEC	0.348	0.347	0.321	0.352	0.352	0.300
	DEC (scaled)	0.201	0.201	0.169	0.172	0.172	0.158
6*Doc2Vec	K-means (rand)	0.586	0.586	0.629	0.712	0.712	0.742
	K-means (++)	0.586	0.586	0.630	0.711	0.711	0.741
	Hierarchy Aggl.	0.444	0.444	0.457	0.602	0.602	0.633
	DBSCAN	0.004	0.004	0.000	0.004	0.004	0.000
	DEC	0.600	0.600	0.629	0.734	0.734	0.759
	DEC (scaled)	0.235	0.235	0.220	0.322	0.322	0.279
Topic Modeling			NMI	AMI	ARI		
LDA			0.350	0.350	0.291

In order to put the sensitivity of the feature representation models to test, they have been trained on two different datasets. The initial one is the KIPRIS patent dataset, and the latter is the concatenation of WoS and KIPRIS datasets, resulting in a larger text corpus. As the results in Table 1 suggest, the FastText mean document embedding based methods generally yield better results in terms of NMI, AMI, and ARI when trained on the larger corpus. Further training on the second and less relevant corpus has resulted in higher scores across the metrics in all clustering approaches, except the modified DEC, where the initial smaller corpus has resulted

substantially better, specifically in terms of ARI. Although this is not easy to assert for the weighted mean FastText document embedding, the weighted mean FastText document embedding model has performed comparatively poorly against the simple mean FastText model and we assume this method is not as sound in this task. Hence, the weighted mean FastText method won't be suggested. In comparison, it can be observed that in Doc2Vec, in almost all clustering approaches, the larger corpus has yielded much better results, where the ARI variance reaches 0.13 in the modified DEC. Overall, Doc2Vec has outperformed FastText based embeddings, where the highest ARI for FastText based methods has been achieved using modified DEC and the smaller corpus is 0.504, much lower than the 0.759 in Doc2Vec.

We observed that DBSCAN is unable to reach stable clustering results, while it fails to determine the correct cluster size (k). This result is less than optimal, hence, will not be commented any further. K-means and Hierarchical agglomerative clustering methods are amongst popular methods in the field. Hierarchical agglomerative clustering can be very useful as it can support the decision of the number of clusters, based on a hierarchical dendrogram for clusters. However, in this task, it was observed that the clustering performance of this method is outmatched by k-means, where K-means has yielded an ARI of 0.742 and Hierarchical agglomerative clustering has yielded 0.633. However, it's worth mentioning that the performance of K-means can be affected by random initialization. The best clustering match has been achieved using the modified DEC in this study, where the ARI 0.759 has been reached. We also observe that further normalization of the inputs in the modified DEC method has dropped the performance significantly, denoted by "DEC (scaled)" in Table 1.

In addition to Doc2Vec and FastText, pre-trained BERT embeddings were also put to test to observe the clustering task performance of the BERT models, as the state of the art models. However, we observe that not fine-tuning or further training these models has resulted in lower than expected clustering performance in this study. As observed in Table 2, the best result can be achieved using BERT uncased A12 pre-trained model. While SciBERT is unable to achieve 0.2 ARI. As the SciBERT model has been trained on full document corpora instead of abstracts or short texts, this behavior can be expected and explained. BERT pre-trained embeddings are regarded as general language representations (Devlin et al., 2018). Hence, it can be explained that BERT and similar representations comprise various dimensions that generally make these embeddings very competitive for various tasks, specifically supervised tasks where the representations can be exploited. On the contrary, these representations are more complex and therefore less suitable for clustering tasks, or in general for some specific tasks. In this study where the objective is to cluster documents based on their topics, it is understandable that the

Research Paper

utilization of such general embeddings may fall short. Hence, feature extraction for better task-specific representations should be preferred in such tasks. All in all, it can be concluded that the shortcomings in the BERT models for this task do not only derive from the number of features (or dimensions of vectors) and difficulty of clustering, but also from the complexity, or in this case the generalization of the representations from the (pre)training corpora. Even though the results have been less than optimal, clustering algorithms are showing consistent behavior. DEC still shows superiority against K-means and Hierarchical agglomerative clustering, while DBSCAN cannot perform any sort of clustering on vectors.

Table 2. Comparison of clustering techniques for pre-trained BERT embeddings.

Embedding	Clustering	Evaluation Metrics		
		NMI	AMI	ARI
BERT (uncased A12)	kmeans (rand)	0.19507	0.1948	0.1442
	kmeans++	0.1950	0.1948	0.1442
	Agglomerative	0.2158	0.2156	0.1877
	DBSCAN	0.0042	0.0037	0.0001
	DEC	0.2568	0.2566	0.2377
SciBERT	kmeans (rand)	0.1498	0.1496	0.1266
	kmeans++	0.1492	0.1489	0.1266
	Agglomerative	0.1903	0.1901	0.1505
	DBSCAN	0.0042	0.0037	0.0001
	DEC	0.1776	0.1774	0.1731

All in all, we can see that the combination of major embedding techniques (i.e. FastText mean and Doc2Vec) and clustering techniques (i.e. K-means, Hierarchical, DEC) can outperform the baseline LDA topic modeling, with Doc2Vec incorporated with the Doc2Vec modified DEC outperforming the rest. Also, having a larger corpus to train the Doc2Vec model results in better performance, whereas the same cannot be applied to FastText average models.

5 Case Study: detecting the sub-fields of Artificial Intelligence field of study

As concluded in the former section, the method of Doc2Vec+modified DEC can be used for the extraction of clusters in textual data, representing the fields / sub-fields. Hence, in this study, we used the chosen method to analyse the field of Artificial Intelligence (AI) extracting its sub-fields.

5.1 Case study: Data collection

The data for this case study has been collected from three mainstream journals until the end of 2019: “Artificial Intelligence”, “Artificial Intelligence Review”,

and “Journal of Artificial Intelligence Research”. The reason for excluding other journals is to limit or eliminate the bias in some journals towards specific applications (e.g. health) or approaches (e.g. engineering and deep learning) in AI. As observed in the Web of Knowledge master journal list categories, these three journals were selected to best fit the purpose of this research, having the minimum bias to specific applications or approaches. As illustrated in Figure 5, the data records prior to 1990 are scarce. Hence, the case study considers 3960 publications between 1990 and 2019.

Figure 5. Data growth from 1970 to 2019 in the three journals, yielding over about 4300 records.

The data cleaning stage for this dataset includes the following steps: Initially, duplicated records are removed based on Digital Object Identifiers and titles. Later, records with missing abstracts or titles are removed. Finally, the pre-processing procedure described in Section 3.1 is carried out for textual data cleanup.

5.2 Embedding and clustering

After preparing the data, the data is used to train a Doc2Vec model with the parameters in the previous section. As the results section concluded, the target text corpus helped the clustering and embedding performance. Hence, the Doc2Vec model is trained on the concatenation of the case data and the data in Section 4.1. Then, the document vectors for the case data have been acquired using the trained Doc2Vec model.

The clustering approach in this case study is also identical to the proposed and the winner method in Section 4.3, which is modified DEC. However, in this case study, we do not have access to the ground truth labels. Consequently, the number of clusters (k) is unknown for the case data. Finding the proper number of clusters is very crucial for the clustering task, as the successful models in this study (i.e. K-means and modified DEC) rely on this input variable. Multiple methods are available to address this issue. The well-known methods are the utilization of distortion score with elbow method, silhouette scores, and using a dendrogram.

Research Paper

Initially, we compute the dendrogram for the vectors, to estimate the number of clusters. We use the longest distance among clusters as a cut-off line in our dendrogram. However, to reinforce the selection, silhouette scores are also computed for all K values in the range of 2 to 50. Based on this score and the approximate dendrogram cut-off values, we select the number of clusters. However, it must be elaborated the larger K values will force the clustering to make smaller clusters, sometimes extracting smaller topics, but less accurate as the silhouette scores suggest. In this case study, based on the silhouette score and the dendrogram, 10 clusters are selected. The result of the clustering is visualized as in Figure 6. The results illustrate the number of articles in each category or cluster for each year for the period of 30 years from 1990 to 2019. As seen in the Figure 6, cluster “c 1” is regaining popularity after a decrease in the number of articles in 2000–2010, while having the most number of articles in 1990–2000. Similarly, cluster “c 2” does not have as many articles as it used to have in 90’s. It can also be noticed that for the past decade, “c 6” and “c 7” clusters have been on a steady rise in the number of articles, while “c 6” obviously comes to existence in AI from zero articles from the mid 90’s.

Figure 6. Diagram of clusters in the study period. Y axis is the number of articles (symmetrical). X axis is years.

Further interpretation of these results requires us to tag or label the clusters. An informative way of labeling the results is to use source titles. However, as the source titles in this case study hint at AI, it will not be as useful. An alternative way to tag the clusters is the keyword extraction method, where the most important key phrases from each cluster will be extracted to hint at the context. In the literature, keyword extraction has been done in various ways, including using “RAKE” (Krenn &

Zeilinger, 2020; Rose et al., 2010), “TF-IDF” (Li, Fan, & Zhang, 2007; Vahidnia et al., 2020), or using embedding and similarity measures (Curiskis et al., 2020).

Figure 7. N-gram dictionary histogram of the dataset. Y-axis is frequency and X-axis is N.

In order to prepare the data for tagging and phrase selection, the n _grams in the text body are concatenated based on the taxonomy generated from author keywords in the dataset, by replacing spaces with underscores (artificial intelligence $\rightarrow$ artificial_intelligence). This taxonomy only contains the 95 percentile of keywords n -gram keywords, to cover the most important keywords. Hence, n -gram keywords with a frequency of at least six are kept and the rest are ignored. In addition, a condition of $M > 2N$ has been maintained to keep the keywords with N -grams and M characters. This eliminates the keywords with characters counts lower than $2N$, which usually are generic words and potentially harmful for the data and the text corpus. For this purpose, N -grams are sorted from higher N to lower N , then replaced in the corpus with corresponding words. In this study, $N \in \{1, \dots, 6\}$, as numbers over 6 are usually either errors or very sparse (please refer to Figure 7 for the histogram of N in n -grams). To eliminate any chance of mid-word overwriting, all searches are done by leading and ending spaces, and to make the corpus suitable for this, a leading and ending space is added to all data records. Ranked key phrases are then extracted from the prepared corpus, using a modified TF-IDF method, as in (Vahidnia et al., 2020). This method is known to be simple and works for all sizes of documents. To better interpret the scores, and select the important words, each row is normalized and the values below 0.5 are cut off. The resulting table of context for the clusters has been provided at Table 3.

Although the context provided in Table 3 is not perfect, it is now much easier to interpret and understand the clusters and the evolution of topics. For instance, the

Research Paper

cluster “c 6” hints at the economical context, and the rise in the number of articles in this cluster hints at the appearance of a multi-disciplinary category of studies in the mid 90’s, which has been on the rise since then. Similarly, the rising number of articles in cluster “c 7”, which has a context of machine learning, demonstrates the expected importance of machine learning techniques in AI. In another instance, cluster “c 8”, which can be labeled as knowledge representation, shows a steady growth from mid 90’s until 2016, then it declines at 2017 and 2018. Cluster “c 9”, that can be interpreted as game theory and decision theory, have always had a steady and constant number of publications assigned to.

The proposed method can show changes in a field, including birth, death, growth, and decline. However, the limitation of the current method is the absence of capability to observe the evolution in detail; Observation of the merging and splitting of the clusters would enhance the ability to see how this evolution is happening, and which fields are giving birth to new fields and how the fields are forming.

Table 3. Ordered key phrases of the clusters.

Cluster #	Terms (Normalized TF-IDF score)
c 1	creativity(1.00), sentiment_analysis(0.85), university(0.81), facial(0.79), insect(0.74), dreyfus(0.71), expert_system(0.67), music(0.65), indian_language(0.64), recommendation(0.63), argumentation(0.62), swarm(0.62), data_mining(0.61), face_recognition(0.61), natural_language_processing(0.60)
c 2	ois(1.00), execution(0.98), sinix(0.88), perception(0.80), people(0.75), unix(0.69), team(0.66), discourse(0.62), intention(0.57)
c 3	revision(1.00), contraction(0.70), postulate(0.65), horn(0.65)
c 4	csp(1.00), propagation(0.80), arc_consistency(0.75), backjumping(0.59)
c 5	description_logic(1.00), deep_learning(0.89), ontology(0.74), rcc(0.56)
c 6	auction(1.00), equilibrium(0.74), election(0.66), coalition(0.66), bargaining(0.56)
c 7	support_vector_machine(1.00), classifier(0.68), knee(0.66)
c 8	document(1.00), wikipedia(0.99), wordnet(0.68), dictionary(0.63)
c 9	phase_transition(1.00), minimax(0.89), voting(0.87), alpha_beta(0.75), chess(0.69), backbone(0.64), optimal_solution(0.63), heuristic_function(0.63), game_tree(0.61), ratio(0.59), heuristic_search(0.59), monte_carlo_tree_search(0.55)
c 10	execution(1.00), reward(0.80), ebl(0.77), pomdp(0.68), team(0.66), heuristic_search(0.64), action_model(0.63), portfolio(0.60), monte_carlo_tree_search(0.59), mdp(0.59), conformant(0.58), mdps(0.57)

6 Conclusion and discussion

In this study, we explored various methods for extracting scientific fields / sub-fields from the titles and abstracts of scientific articles. We showed that the modified DEC in conjunction with Doc2Vec can outperform other methods in our clustering task, including FastText and pre-trained BERT with K-means and DEC clustering, LDA, and Doc2Vec with K-means clustering. As reported in the results, the proposed method achieves an ARI of 0.759, while LDA as a baseline and pre-trained BERT

stay below 0.3. We also show that the deep clustering method can outperform K-means and other popular clustering methods, regardless of the embedding method. The inadequate performance of LDA is expected as compared to more advanced methods. However, the weak performance of BERT models may come as a surprise. The utilization of pre-trained models, and not fine-tuning or further training of the models do impact the results, especially in the task of this study. The task of the study, which is clustering as a means of extraction of topics, requires the models to be trained on a corpus of similar text and concepts, making the embedding models more aware of the context of the test data. Hence, the inferior performance in clustering for topic detection of the BERT models can be explained by the use of general-purpose models, the complexity of the features, and its dimensions. The higher number of dimensions can mean a higher number of learned features. But it is only useful if trained on a relevant dataset with relevant objectives. Additionally, it must be mentioned that for a clustering task, having too many dimensions is generally not desired. Consequently, it can be concluded that pre-trained BERT does not necessarily result in superior results.

Using the proposed method on the case data of AI publications reveals the topics and interdisciplinary fields in AI, and their appearance and disappearance throughout the years 1990 to 2019. We provide a simple visualization method for the popularity of studies for each year and topic. The visualization is simple to use and easy to replicate, and provides necessary information to observe the state of topics in each year. However, a limitation of this visualization is the lack of ability to demonstrate the merging and splitting of the fields. A way to overcome this limitation is to use Sankey diagram, similar to the prior study (Vahidnia et al., 2020). We also show the effectiveness of the modified TF-IDF as a keyword extraction method in giving context to the clusters and providing a way to interpret the cluster, or to be used as a cluster labeling method.

It should be discussed that the some interpretations can be subjective and quantitative performance evaluation is not possible for the case study. As explained in Section 3.4, DEC and k-means are not invariant to randomness, and as a result, any two runs may result in either slight or major variations in the clustering results. Therefore, in this article, for a fair comparison, the best results in multiple runs have been reported. Although in this study an attempt has been made to test the effect of variance in data by using two different corpora for training, due to limitations it was not possible to assess the effect of corpus size and diversity for clustering extensively.

In future works, it is planned to use transformers as the embedding methods and utilize network properties of the documents to explore the more novel document representations models, potentially enhancing the overall performance. Additionally, taking advantage of soft clustering approaches will be explored in future studies.

Author contributions

Sahand Vahidnia (s.vahidnia@unsw.edu.au): Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, and Writing (original draft, and review and editing); Alireza Abbasi: Supervision, Conceptualization, Formal analysis, Methodology, and Writing (review and editing); Hussein A. Abbass: Supervision, Conceptualization, Methodology, and Writing (review and editing).

References

- Arora, S., Liang, Y.Y., & Ma, T.Y. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *proceedings of International Conference on Learning Representations*, Toulon, France, April 24–26, 2017.
- Astrakhantsev, N. (2015). *Methods and software for terminology extraction from domain-specific text collection* (Unpublished doctoral dissertation). Ph. D. thesis, Institute for System Programming of Russian Academy of Sciences.
- Awan, M.N., & Beg, M.O. (2020). Top-rank: A topical position rank for extraction and classification of keyphrases in text. *Computer Speech & Language*, 65, 101116.
- Beltagy, I., Lo, K., & Cohan, A. (2019). Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.
- Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.
- Cagliero, L., & La Quatra, M. (2020). Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications*, 160, 113659.
- Curiskis, S.A., Drake, B., Osborn, T.R., & Kennedy, P.J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, 57(2), 102034.
- Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391–407.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).
- Harris, Z.S. (1954). Distributional structure. *Word*, 10(2–3), 146–162.
- Hou, J.H., Yang, X.C., & Chen, C.M. (2018). Emerging trends and new developments in information science: A document co-citation analysis (2009–2016). *Scientometrics*, 115(2), 869–892.
- Jelodar, H., Wang, Y.L., Yuan, C., Feng, X., Jiang, X.H., Li, Y.C., & Zhao, L. (2019). Latent dirichlet allocation (lda) and topic modeling: Models, applications, a survey. *Multimedia Tools and Applications*, 78(11), 15169–15211.
- Jones, K.S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 11–21.

- Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
- Kenter, T., Borisov, A., & De Rijke, M. (2016). Siamese cbow: Optimizing word embeddings for sentence representations. arXiv preprint arXiv:1606.04640.
- Kim, J., Yoon, J., Park, E., & Choi, S. (2020). Patent document clustering with deep embeddings. *Scientometrics*, 1–15.
- Krenn, M., & Zeilinger, A. (2020). Predicting research trends with semantic and neural networks with an application in quantum physics. *Proceedings of the National Academy of Sciences*, 117(4), 1910–1916.
- Kuhn, T., Perc, M., & Helbing, D. (2014). Inheritance patterns in citation networks reveal scientific memes. *Physical Review X*, 4(4), 041036.
- Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).
- Li, J.Z., Fan, Q.N., & Zhang, K., et al. (2007). Keyword extraction based on tf/idf for chinese news document. *Wuhan University Journal of Natural Sciences*, 12(5), 917–921.
- Liu, H.W., Kou, H.Z., Yan, C., & Qi, L.Y. (2019). Link prediction in paper citation network to construct paper correlation graph. *EURASIP Journal on Wireless Communications and Networking*, 2019(1), 1–12.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
- Miller, G.A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11), 39–41.
- Peganova, I., Rebroya, A., & Nedumov, Y. (2019). Labelling hierarchical clusters of scientific articles. In *2019 ivannikov memorial workshop (ivmem)* (pp. 26–32).
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
- Radu, R.-G., Rădulescu, I.-M., Truică, C.-O., Apostol, E.-S., & Mocanu, M. (2020). Clustering documents using the document to vector model for dimensionality reduction. In *2020 IEEE international conference on automation, quality and testing, robotics (aqtr)* (pp. 1–6).
- Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: Applications and theory*, 1, 1–20.
- Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 53–65.
- Steinley, D. (2004). Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3), 386.
- Vahidnia, S., Abbasi, A., & Abbass, H.A. (2020). Document clustering and labeling for research trend extraction and evolution mapping. In C. Zhang, P. Mayr, W. Lu, & Y. Zhang (Eds.), *Proceedings of the 1st workshop on extraction and evaluation of knowledge entities from scientific documents co-located with the ACM/IEEE joint conference on digital libraries in 2020, eek@jcdl 2020, virtual event, china, august 1st, 2020* (Vol. 2658, pp. 54–62). Retrieved from <http://ceur-ws.org/Vol-2658/paper7.pdf>
- Ward Jr, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301), 236–244.

Research Paper

- Weber, T., Kranzlmüller, D., Fromm, M., & Tavares de Sousa, N. (2020). Using supervised learning to classify metadata of research data by field of study. *Quantitative Science Studies*, 1–26.
- Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478–487).
- Xu, H.Y., Winnink, J., Yue, Z.H., Liu, Z.Q., & Yuan, G.T. (2020). Topic-linked innovation paths in science and technology. *Journal of Informetrics*, 14(2), 101014.
- Xu, S., Hao, L.Y., An, X., Yang, G.C., & Wang, F.F. (2019). Emerging research topics detection with multiple machine learning models. *Journal of Informetrics*, 13(4), 100983.
- Xu, S., Zhai, D.S., Wang, F.F., An, X., Pang, H.S., & Sun, Y.R. (2019). A novel method for topic linkages between scientific publications and patents. *Journal of the Association for Information Science and Technology*, 70(9), 1026–1042.
- Zeng, A., Shen, Z.S., Zhou, J.L., Wu, J.S., Fan, Y., Wang, Y.G., & Stanley, H.E. (2017). The science of science: From the perspective of complex systems. *Physics Reports*, 714–715, 1–73. Retrieved from <https://doi.org/10.1016/j.physrep.2017.10.001> doi: 10.1016/j.physrep.2017.10.001
- Zhang, Q.R., Li, Y., Liu, J.S., Chen, Y.D., & Chai, L.H. (2017). A dynamic co-word network-related approach on the evolution of China’s urbanization research. *Scientometrics*, 111(3), 1623–1642. doi: 10.1007/s11192-017-2314-1
- Zhang, Y., Chen, H.S., Lu, J., & Zhang, G.Q. (2017). Detecting and predicting the topic change of knowledge-based systems: A topic-based bibliometric analysis from 1991 to 2016. *Knowledge-Based Systems*, 133, 255–268. Retrieved from <http://dx.doi.org/10.1016/j.knosys.2017.07.011> doi: 10.1016/j.knosys.2017.07.011
- Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H.S., & Zhang, G.Q. (2018). Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics*, 12(4), 1099–1117.
- Zhang, Y., Zhang, G.Q., Zhu, D.H., & Lu, J. (2017). Scientific evolutionary pathways: Identifying and visualizing relationships for scientific topics. *Journal of the Association for Information Science and Technology*, 68(8), 1925–1939. Retrieved from <http://doi.wiley.com/10.1002/asi.23814> doi: 10.1002/asi.23814
- Zhou, Y., Lin, H., Liu, Y.F., & Ding, W. (2019). A novel method to identify emerging technologies using a semi-supervised topic clustering model: A case of 3d printing industry. *Scientometrics*, 120(1), 167–185.

This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).