Scientometrics (2022) 127:5385–5414
https://doi.org/10.1007/s11192-021-04187-4

Doc2vec-based link prediction approach using SAO structures: application to patent network

Byungun Yoon[1] · Songhee Kim[1] · Sunhye Kim[1] · Hyeonju Seol[2]

Received: 27 February 2021 / Accepted: 11 October 2021 / Published online: 13 November 2021

Abstract
As the amount of documents has exploded in the Internet era, many researchers have tried to understand the relationships between documents and predict the links between similar but unconnected documents. However, existing link prediction techniques that use the predefined links of documents might provide incorrect results, because of the generic problem of citation analysis. Moreover, they may fail to reflect important contents of documents in the link prediction process. Thus, we propose a new link prediction approach that employs the Doc2vec algorithm, a document-embedding method, in order to predict potential links between documents, by reflecting the functional context of technological words. For this, first, we collected both citation information and documents of patents of interest, and generated a patent network by using the citation relationship between patents. Second, we identified unconnected links between nodes and transformed the patent document into document vectors, based on the Doc2vec algorithm. In particular, since patent documents include useful functions for solving technological problems, the proposed approach extracts subject-action-object (SAO) structures that we used to generate document vectors. Then, we calculated the similarity between patents in the unconnected links of a patent network, and could predict potential links by using the similarity. Third, we validated the results of the proposed approach by comparing them using the Adamic–Adar technique, one of the traditional link prediction techniques, and word vector-based link prediction. We applied the Doc2vec-based link prediction approach to a real case, the unmanned aerial vehicle (UAV) technology field. We found that the proposed approach makes better predictions performance than the Adamic–Adar technique and the word vector approach. Our results can help analyzers accurately forecast future relationships between nodes in a network, and give R&D managers insightful information on the future direction of technological development by using a patent network.

Keywords Link prediction · Patent network · Doc2vec · Document embedding · Unmanned aerial vehicle

✉ Hyeonju Seol
hjseol@cnu.ac.kr

[1] Department of Industrial & Systems Engineering, Dongguk University, Seoul 04620, South Korea

[2] School of Integrated National Security, Chungnam National University, Daejeon 34134, South Korea

Introduction

As advanced Internet and computing technology has developed and social network services have generated many social relationships, our society has become complexly connected. From the inception of human history, human networks have existed, because humans instinctively want to live in a connected world. Since social network services have rapidly proliferated online, social relationships among human have become complex and intensified. Such networks can include a wide spectrum of relationships, including friendships, affiliations, and collaborations. The forms of networks can be physical (off-line) or non-physical (on-line) in the Internet era. Although the relationships among actors are fundamental and useful in network analysis, it is valuable to find new relationships that have not been connected. Link prediction is a data mining technique to predict the presence and absence of links between disconnected nodes (Wu et al., 2017). Such approaches enable the recommendation of new friends in social network services, and potential partners in a collaborative network of firms (Hopcroft et al., 2011; Tang et al., 2012). Since an actor in a network generally has limited information and relationships, he or she might be unable to search for latent relationships among the weak links of a network. Thus, link prediction that finds out potential links can support a process to increase connectivity in a network.

Many researchers have suggested various methods for improving the performance of link prediction including metric-based approaches and learning-based approaches. The metric-based approaches, such as the Adamic-Adar index and Jaccard index, measure the possibility of a connection between unconnected nodes by calculating their relationships with specific indices (Liben-Nowell & Kleinberg, 2007; Liu & Lü, 2010). In contrast, the learning-based approaches mostly use machine learning, such as a support vector machine and random forest to predict potential links. The main problem of traditional approaches is to deal with only the relationships among nodes in a network. If one has only the information on the links in a network, existing approaches might provide a solution. However, if a node in a network is a document, its contents can be considered in order to analyze the relationships between documents, rather than citing–cited relationships. Thus, traditional approaches for link prediction have several limitations. First, the information on existing links is produced by individuals in various cases, such as social network services and references to academic papers. Thus, it is very subjective, because individuals judge such links within the scope of their own knowledge, hence may overlook real relationships. Second, the main approaches for link prediction generally depend on obvious links that exist in a network. Although these are an important source for prediction of potential links, textual information in a document network can be used for making precise and profound link predictions. Recently, several studies have suggested content-based approaches for investigating textual information in documents in link prediction. However, since such approaches consider almost only words or word vectors in analyzing the relationships between documents, they mostly do not work well in reflecting contextual information of documents. In particular, when link prediction techniques are applied to patent documents, general approaches may overlook the characteristics of technological information that focuses on technological functions. Third, since existing link prediction models mostly use graph-based similarity metrics, they focus on analyzing the indirect relationships between nodes in a network. For example, the Jaccard coefficient and Adamic-Adar index consider the number of common neighbors in order to predict potential links. Their main idea is that having many common neighbors makes it likely that two nodes can be connected. However, the number of common neighbors is indirect information, and we can easily notice that there are many pairs of people who have a lot of common friends but

are not themselves close. If two nodes have many common neighbors but are not connected, these two nodes might be unconnected for a reason, and we might anticipate that they would remain unconnected in the future. Fortunately, the direct relationship between documents can be analyzed in a document network for link prediction because we can calculate the similarity between documents based on text information. Thus, although most of the existing approaches used an indirect relationship in a generated network, a network of documents requires a new approach for link prediction by using textual information.

In this paper, we propose a new approach for predicting a potential link in a network by considering the contents of documents. Thus, the proposed approach can be applied to a network that has textual information, including networks of academic papers and patent networks. For this, we used text mining techniques and Doc2vec, a kind of word2vec extension, to analyze the relationships between documents. Generally, for a contents-based approach, the similarity of keywords that are included in two documents is useful as a simple technique for exploring potential links in a network that are not connected. However, since language is often complex and abstruse, the context in which a word is used should be considered in order to grasp the intent of the writer or speaker. Thus, we applied the Doc2vec technique, rather than simple keyword similarity, in order to investigate the accurate meaning of authors. The approach can predict potential links among nodes with more accuracy. Such a context-based approach can provide a view different from that of existing link prediction approaches, and consider latent relationships between documents that do not have any connections, such as references and hyperlinks. In predicting the links in patent networks, textual information will be more useful for finding a potential link that can be connected than is citation information between patent documents. Technological contents in patent documents can provide more accurate information than can citing–cited relationships for link prediction. In particular, since technological words have more specified meanings for a technology than do ordinary words, a word-embedding technique can generate a plausible patent network. In addition, patent documents include many subject-action-object (SAO) structures, because they need to describe specific solutions to technological problems. Thus, our proposed approach employs the Doc2vec algorithm based on SAO structures rather than general sentences of patent documents.

The rest of this paper is organized as follows. "Related work" section reviews the concepts, methods, and applications of link prediction and word embedding. "The proposed approach" section explains the proposed approach for link prediction based on Doc2vec. "Results" section then introduces the case study of patent analysis in the unmanned aerial vehicle (UAV) field. Finally, in "Discussion" section we discuss the contributions and limitations of this study.

Related work

Link prediction

A link is a pattern that exists anywhere in a general relationship, and can represent an attribute of an instance, such as the importance, rank, or category of an entity (Taskar et al., 2004). It allows analysts to predict how future links will appear, based on previously observed links in various areas that have evolved over time (Getoor & Diehl, 2005; Zhang et al., 2018). That is, it is a method to estimate the possibility that a link exists between two nodes, considering the attributes of links and nodes (Al Hasan et al, 2006; Lü & Zhou, 2011). Although there may be no association between nodes at present, predicting future

associations between two nodes is possible. A network is a structure in which nodes can represent people or other entities embedded in various contexts, and edges represent interactions, collaborations, or influences between entities. Therefore, link prediction, which aims to anticipate potential interactions in a network model, has been an important issue in network analysis and has recently been studied in various contexts, such as social networks, genetic interaction networks, and literature-citation networks (Chen et al., 2005; Getoor & Diehl, 2005). In particular, although link prediction is a complex problem in understanding dynamics, because there are diverse variables in the dynamics that lead to the evolution of the network, this approach started with the idea that it is relatively simple to compare two specific nodes for link prediction (Al Hasan et al., 2006; Behrouzi et al., 2020).

Link prediction approaches can be classified into three categories, such as link-based, content-based, and hybrid approaches, in terms of data for analysis. The link-based approaches, such as the Adamic-Adar index, Jaccard index, and common-neighbor index, formulate an index of links between nodes for link prediction. They generally try to find a pair of unconnected nodes that are close in a network. In contrast, content-based approaches analyze the profiles of nodes. In particular, if a node is a document in a network of documents, keywords or terms are very useful for investigating the similarity between documents. The hybrid approaches combine both links and contents for link prediction. Since they require much data and time for analysis, the use of hybrid approaches is limited to specific cases that have a sufficient database. In addition, the classification of link prediction approaches can be done in terms of methodology. In general, metric-based approaches construct a specific metric to find missing links between nodes. Thus, this type can be applied to contents-based approaches as well as to link-based approaches. Although many metrics, such as the Adamic-Adar index and Jaccard index, have been proposed to predict potential links in the link-based approaches, the similarity metrics can be applied to analyze the relationships of nodes based on their contents. In contrast, learning-based approaches use learning models that apply patterns of connected nodes to unconnected nodes in order to identify potential links. The support vector machine (SVM) and random forest are representative methods that are employed for link prediction.

Recently, interest in computer analysis of networks has been increasing, because of the importance of large and complex networks and their properties (Liben-Nowell & Kleinberg, 2007; Popescul & Ungar, 2003). At the inception of the study area, algorithms based on Markov chains and various statistical models were proposed in the computer science community (Lü & Zhou, 2011). Previous studies of link prediction include pattern recognition in the dataset, described as a set of independent instances of a single relationship found by conventional data mining algorithms, such as association rule mining, market basket analysis, and cluster analysis for link mining, and labeling web pages based on the characteristics of neighbors connected with the features of the web page (Adamic & Adar, 2003; Getoor, 2003; Getoor & Diehl, 2005).

Link prediction approaches have been actively applied to many cases where networks can be generated (Jeong et al., 2021). For example, potential collaborators can be found by using link prediction techniques, because missing links between people or companies can be identified in a collaborative network. Collaboration between researchers is often synergistic, and organizing such a team is difficult. Although individual researchers can achieve better results in collaboration, they are often unaware of the existence of other researchers. Hence it is unclear where potential collaborators are or how such collaborations should be built (Pavlov & Ichise, 2007). However, if a researcher recognizes that the other researcher has conducted similar research, the researcher will be able to find a collaborator who has expertise in the field. Since predicting the likelihood of collaborative work is conceptually

and structurally similar to the actual social network problem, it would appear to be an important research direction (Al Hasan et al, 2006; Huang et al., 2004). Thus, link prediction can show potential collaborators in a network of academic papers or patents.

In this paper, we try to estimate the possibility of the existence of links between nodes that do not have links based on existing links. Based on the similarity between documents (distance based on Doc2vec), we suggest a new approach that recommends a patent document with high similarity based on SAO structures, even though there is no citation relation.

Document representation

Word embedding, one of the representative NLP techniques, is a method of translating the meaning of a text into a machine-understandable format by digitizing a word constituting a text, and mapping the word in a vector space (Goldberg & Levy, 2014; Levy & Goldberg, 2014; Rajbabu et al., 2018). Likewise, document embedding is also applied to paragraphs or documents (Dai et al., 2015; Lau & Baldwin, 2016). Although the one-hot vector method was used in the early days of the method for vectorizing text, various other methods were developed, because only the information on the occurrence of the word can be provided in the method, leading to large information loss (Rong, 2014; Turian et al., 2010). Currently, many researchers use a count-based method based on identifying the number of occurrences of a word, and a prediction-based method based on word prediction. Count-based methods include count vectors and Tf-idf, whereas prediction-based methods include word2vec, which has two different approaches, namely, *Continuous Bag of Words* (CBOW; current word prediction using context), and *Skip-gram* (neighboring word prediction with current word) (Le & Mikolov, 2014; Mikolov et al., 2013a, 2013b). Although the method of vectorizing the document initially used the one-hot vector method, the Doc2vec or paragraph2vec algorithm is currently being used as an extension of the word2vec method proposed by Mikolov.

By using word and document embedding, it is possible to do quantitative analysis (classification, clustering, etc.), which is an essential condition for machine learning, and to process it quickly, by transforming unstructured text data into quantitative data (Tang et al., 2015; Li et al., 2016). Word embedding and document embedding are used to capture semantic and syntactic information of words, and to measure the similarity of words that are widely used in various information retrieval and natural language processing tasks (Xie et al., 2020). Document embedding has shown that there are many similar documents in a vector space with close vectors of documents, so it can be used for various fields, including movie review classification, web page search, and sentiment classification (Le & Mikolov, 2014; Liu et al., 2015; Tang et al., 2014). Based on the similarity measure, it can be applied to diverse fields, such as information protection and document classification (Huang et al., 2012). Mikolov showed that grouping similar words or phrases by means of vectorization of words and sentences could work better for natural language processing, and could then be applied to various studies, such as the analysis of future technology (Mikolov et al., 2013a, 2013b). After Doc2vec, many document- or sentence-embedding models, such as FastSent and Sentence-bert that embed documents have been proposed. However, document embedding using the Doc2vec model is still widely used, because of its fast learning speed and high efficiency. In addition, the document vector resulting from Doc2vec is a vector obtained based on the co-occurrence frequency of words that appear simultaneously in a certain data window and well represent the order and context of words that appear in the patent document. Therefore, it is an appropriate vector to be used to calculate the similarity in technical contexts between two documents in this study.

When a document is transformed into a document vector, several types of data can be used. In general, keywords have been widely used to generate a document vector by filling the occurrence frequency of keywords in data fields of a vector, since the keyword-based approach may overlook specific relationships between words and the context of words in a specific document (Kroeger, 2005). In particular, since technical documents such as patents and academic papers normally have a lot of technological information, many researchers have analyzed function-oriented contents. Thus, SAO structures that use verbs to describe the functions of technology are useful for investigating the characteristics of technical documents. When analysts perform function analysis, a system is structured by subjects, actions, and objects. In the SAO structure, since the combination of action and object indicates a technical problem and a subject can be regarded as a solution, the relationship between problems and solutions can be inferred (Moehrle et al., 2005). Most existing studies extract the frequency of SAOs and conduct semantic-equivalence calculations based on dictionaries to assess the similarity between patent documents. The SAO structures have been used to satisfy various needs such as trend analysis, technology tree analysis, and patent network analysis.

Thus, we use the Doc2vec algorithm that extends word2vec, which learns words using a neural network based on the number of occurrences of words together. Using the above algorithm, we measured the similarity between documents by converting SAOs of patent documents into numerical data. We expect that patent documents that do not have a citation relationship, but have similar research topics, can be identified and used to recommend potential technology convergence.

The proposed approach

Research concept

In general, link prediction requires a network that is composed of nodes and links, in order to explore new links between nodes. In a document network, the citing–cited relationship is basic information needed to make the link prediction. Since such a network is developed within the knowledge of authors, it might not consider actual relationships between nodes, leading to inaccurate link prediction. Thus, in this paper, we used the similarity of documents based on content-based features to predict potential links. In addition, the context in which SAOs are used in documents should be considered, to investigate more accurate relationships between documents. Doc2vec can deal with this issue by analyzing how an SAO is used in a document with other SAOs and generating document vectors. Although a network is constructed by citation information that patent documents have, missing links that can be connected between unconnected pairs of nodes can be explored by means of context-oriented similarity, based on Doc2vec and SAO structures. If a vector similarity of a pair of patent documents is higher than a cut-off value, the proposed approach can connect them. Newly connected pairs of patents can be candidates for technology convergence, because, despite their strong relationship, they have not previously been linked. Figure 1 shows the conceptual scheme of Doc2vec-based link prediction.

Overall process

Since the proposed approach deals with the analysis of textual data by means of machine learning, the overall process consists of multiple steps to derive the intended outputs. First,

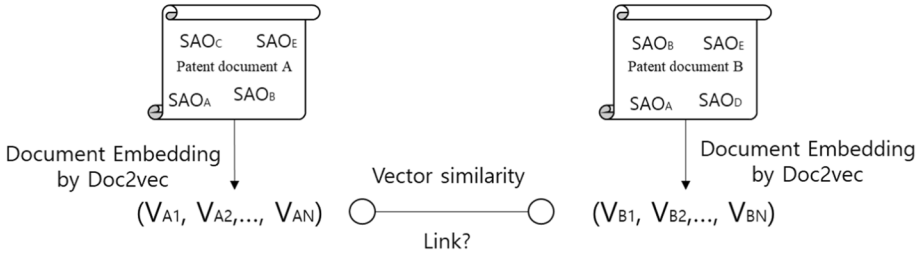

Fig. 1 Conceptual scheme of doc2vec-based link prediction

we selected the technology field of interest and did the data collection. The dataset should include citation information and textual information. Although there are many national patent databases, we chose the USPTO, which is a bigger database than are the main databases of several countries, such as the EU, Japan, and Korea. Second, we drew a patent network by visualizing the citing–cited relationships of patents. Thus, the nodes of a patent network are patents, and the links between patents are the existence of citation relationships. When a patent cites another patent, the two patents are linked. Third, patent documents of which the patents are included in a developed network are, by means of the Doc2vec technique, transformed into document vectors. The document vectors of patents consist of the value of features that provide the characteristics of patents. For this, we extracted SAO structures by analyzing syntactic information of sentences and used them to generate document vectors by means of the Doc2vec algorithm. Fourth, we predicted the potential links that can be connected between unconnected links by using the derived document vectors of patents. We used the similarity between patents to find pairs of patents that have high similarity, but are not yet connected. Finally, to validate its value, we compared the performance of the proposed approach with that of existing methods. Figure 2 presents the overall process of the proposed approach for link prediction.

Constructing a patent network

For link prediction, a patent network that consists of nodes and links should be drawn by visualizing the relationships between patents. Whereas nodes in a patent network refer to patents that are collected from a patent database, links between patents can be assigned by investigating whether a citing–cited relation between patents exists or not. Patent documents have references that show the list of patents or academic papers referred to, to develop the patent, or write the patent document. Thus, we automatically extract the citation relationships from patent documents to generate a patent network.

In terms of types of network, a patent network is a directed network, because two patents cannot cite each other. Since patents are granted in temporal order, a patent should cite already assigned patents, leading to the construction of a directed network. A node (patent) in a network can have multiple links, because a patent has more than one reference. If patent A cites patent B, the arrow of a link in a patent network starts from patent A and ends in patent B.

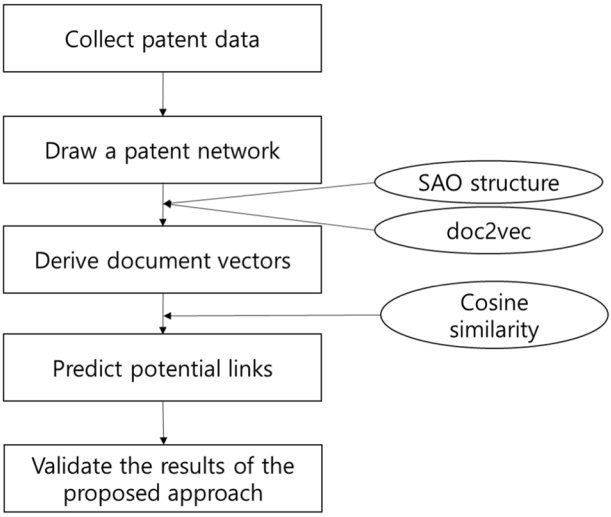

Fig. 2 Overall process of the proposed approach

Preprocessing by means of text mining

In order to predict missing links in a patent network by the proposed approach, patent documents that are collected should be transformed into document vectors. Since a patent document is in a form of natural language that could not be visualized in an intact form, it needs to be changed into a vector form that can be quantitatively analyzed. For this, we can consider two types of feature extraction: keywords and SAOs. If we use keywords to generate keyword vectors of documents, a general process of keyword extraction should be used. First, since many meaningless words are included in a document, stop-words, such as prepositions, articles, and conjunctions ('a', 'the', 'in', 'and', and so on) should be excluded from the list of extracted words. Second, keywords that can explain important features need to be identified by investigating the term frequency–inverse document frequency (tf-idf) value of words. In general, a word that frequently appears in documents is mostly useless for document analysis, because it cannot deliver unique meaning. In addition, if a word scarcely occurs in documents, it also cannot distinguish various documents. Thus, we applied the tf-idf value that weighs each term by dividing term frequency by the number of documents in the corpus containing the term. Third, after extracting important words from a pile of words, we calculated the frequency of keywords and derived the keyword vectors of all collected patents to assess the similarity between patents. The data fields of keyword vectors are filled with the frequency of keywords.

However, we propose the use of SAOs to generate a vector of documents. Thus, a parser that investigates a string of symbols in natural language, conforming to grammar rules, is needed to extract SAOs for a document. Parsers can show semantic relationships of words by analyzing components of sentences. The Stanford Parser has been widely used because it is available as an open-source program (Toutanova, K., & Manning, 2000; Manning et al., 2014; Chen & Manning, 2014). Thus, we employed the Stanford Parser to divide sentences and extract SAO structures from the sentences. By using the Stanford parser, post-tagging of each word and word-to-word relationships can be obtained in a sentence.

Then, using the rule-based model, one can find the subject, verb, and object phrases, and then define the phrases with the verb corresponding to the action as an SAO structure (Guo et al., 2016). For example, if we have a sentence, "A cup holds coffee", the "cup" is the subject because it is doing the action. In addition, whereas the word "holds" is the action that is what the subject does, the "coffee" is the object to be acted on in the sentence. Thus, a document can be characterized by sentences that have an SAO structure.

Predicting potential links based on Doc2vec

In this paper, we carry out link prediction in a patent network by using the Doc2vec algorithm and SAO structures. For this, patent documents need to be broken down into words and sentences that are basic components of documents. Then, SAOs are identified by analyzing the relationships of subjects, actions, and objects. For the sentences that have technical information, SAOs are used as an input for applying the Doc2vec, and they are learned by considering the contexts of sentences. Existing Doc2vec used all sentences and words in the document equally for learning. In this study, not all sentences were used, but only the SAO structure was used for learning. In other words, instead of learning the structure of words in all sentences, only the structure of words in SAO, which means technical problems and solutions, is learned. This results in a document vector that is more descriptive. The Doc2vec algorithm uses an unsupervised learning approach to learn the document representation. Although the input of words per document can differ, the output of this approach is fixed-length vectors. The algorithm in Appendix 1 shows the codes of Doc2vec to generate the document vectors of patents. The Doc2vec algorithm has a PV-DM and a PV-DBOW method that can be visualized as shown in Figs. 3a and b, respectively. Whereas the PV-DM method is similar to word2Vec's CBOW method in predicting the word that will appear after a sequence of words and updating the paragraph vector, the PV-DBOW method predicts words that will appear in a context using only the paragraph vector, as in the skip-gram method. The objective function of Doc2vec is the direction that maximizes the log average probability in a sentence with t words, as shown in Eq. 1, in the same way as Word2Vec. That is, it is learned in the direction of maximizing the probability $P(w_t|w_{t-k}, \ldots, w_{t+k})$, where w_t appears as the target word when the word $w_{t-k},\ldots\ w_{t+k}$ appears at the times $t+k,\ldots\ t+k$. In this study, we used a vector using both the PV-DM method and the PV-DBOW method, as recommended in the study that proposed Doc2vec. More technically, Doc2vec is applied by using 'gensim', a package that implements Doc2vec by means of python. In principle, Doc2vec consists of an input layer, a hidden layer, and an

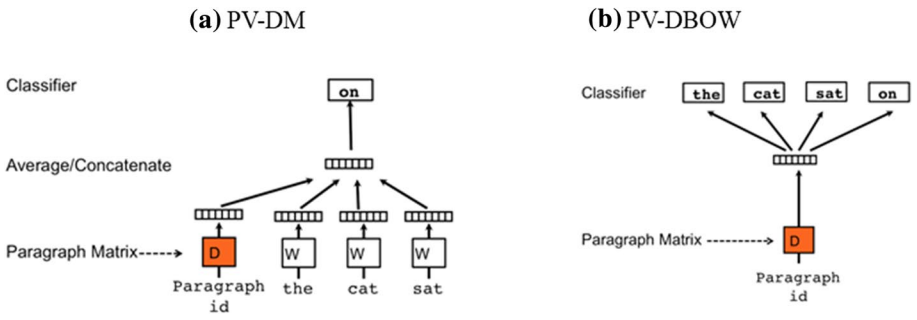

Fig. 3 Visualization of doc2vec algorithms

output layer. Since it is unsupervised learning, there is no target data and the output layer does not need to be defined by an analyst. In this study, the input layer inputs previously defined sentences in which the SAO structure appears. At this time, the words of each sentence are input as one token in document unit. For the hidden layer, we need to set the number of hidden units as a parameter.

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},\ldots,w_{t+k}) \tag{1}$$

Link prediction can be conducted in several ways to get accurate results, according to the characteristics of data. In this paper, we used the patent documents as a critical data source for the link prediction. Although other databases typically have information only on links, patent documents can provide profound information from textual data that can upgrade link relationships with contextual facts. Doc2vec is employed to generate the document vectors of patents by considering the technological contexts that words are used in. The relationships between patent documents are assessed by calculating their similarity, based on the document vectors. Although the similarity can be measured by various methods, we selected the cosine similarity to investigate the potential links between unconnected pairs of patents in an existing patent network. Equation 2 provides the formulation of cosine similarity. If the similarity is higher than a predefined cut-off value, a link between patents should be connected in a patent network. The cut-off value can be quantitatively or qualitatively decided by data analysis or expert judgment. For a quantitative approach, the mean value of the averages of vector similarity in already linked and non-linked groups can be a cut-off value for link prediction. If the similarity of the patent vector in an unconnected link is higher than that of a cut-off vector, the link can be predicted as a potential link that can be connected in the future.

$$\text{cosine similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2}$$

Validating the proposed approach

In this paper, we propose a new approach to predicting a potential link between patents by using the semantic similarity based on Doc2vec. Thus, the results of the proposed approach can be validated by comparison with those of existing link prediction approaches. Although there are many techniques for link prediction, we selected two representative techniques of Adamic–Adar index and semantic similarity based on the frequency of words. First, in order to validate the performance of the proposed approach, we compare it with that of link prediction using a traditional semantic-similarity approach. Since a confusion matrix is commonly used to evaluate the accuracy of prediction, we compare two confusion matrices of the Doc2vec approach and frequency-based approach for link prediction. We used two indices, precision and recall, to compare the results from the confusion matrix. Second, since the Adamic-Adar index is a representative link prediction method and several studies show the effectiveness of this index (Sun et al., 2017; Lü, L. & Zhou, T., 2011), we used the Adamic-Adar index to validate the proposed approach. In general, although it shows good results in predicting the friendship in a human network, it shows a poor accuracy prediction in the experiment of predicting author collaboration (Adamic & Adar, 2003). Thus, we need to apply the Adamic-Adar technique to a patent network and compare the

performance of two approaches. In terms of the Adamic–Adar approach, although all data should be split into two parts (training and test data) in the semantic-similarity approach, this method should be used to analyze the links of all data without split data. If all nodes and links are not included in the link prediction, the prediction is incomplete, by failing to consider all existing links between all nodes. In particular, all relationships of the patent citation should be analyzed in a patent network for link prediction without the exclusion of a patent. Thus, the split of data into training and test data makes it impossible to apply the Adamic–Adar technique to the patent data. Since the confusion matrix cannot be made in the Adamic–Adar technique, we derive the top 20 relationships between patents that can be linked between unconnected patents. In order to validate the performance of the proposed approach, we investigate whether the top 20 links by the Adamic–Adar technique are included in the list of potential links that are drawn by the proposed approach.

Results

Data collection

We selected a state-of-the art technology field to illustrate the application of the proposed approach. Although there are many emerging technology areas, unmanned aerial vehicle (UAV) technology is an innovative area that can dramatically change the industry structure. The UAV, commonly known as a drone, refers to an aircraft without an onboard human pilot, enabling autonomous flight under the remote control of a human or computer, or fully autonomous control by onboard computers. The applications of this technology have been expanded from military uses to commercial, scientific, and other applications. Since the physical structure of the UAV includes sensors, actuators, computers, and energy supply, various technologies should be integrated for stable flight. Thus, many researchers and technologists have developed emerging technologies to realize the applications of the UAV in practice.

In this paper, we use the USPTO patent database, because it has abundant valuable information, such as patent citations and well-structured patent documents. Furthermore, many companies are eager to apply their patents to the USPTO, which, between several national patent organizations, has the largest U.S. market and the highest reputation. In order to collect UAV-related patents from the patent database, an appropriate searching query should be identified. This includes proper keywords, such as unmanned, fly, vehicle, and drone, as shown in Appendix 2. Consequently, we retrieved 1,188 patents from the database, covering the time period until 2019. We split the training and test data randomly in a 7:3 ratio for link prediction.

Development of a patent network

We used patent citation information to generate a patent network that consists of nodes and links. In this paper, we developed the UAV-related patent network by visualizing the citation relationships of 1188 patents with the network analysis software nodexl. Figure 4 shows the patent network of UAV-related patents. In the patent network, there are 1737 nodes, using citing and cited patents because we should include patents that are not collected but cited by collected patents. After the patent network is generated, we can find that the number of groups is 13. In the largest group, the number of nodes is 1445, and the total

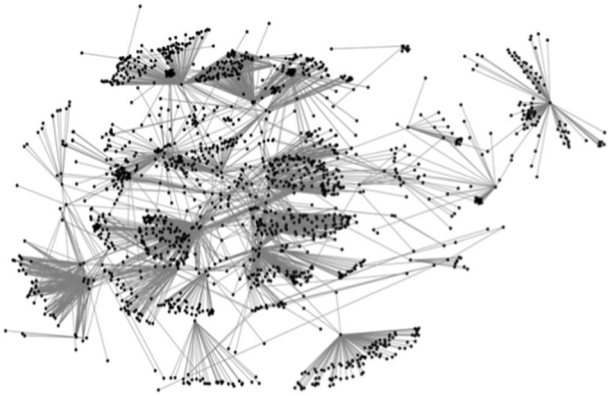

Fig. 4 Patent network of UAV-related patents

number of links is 3118. We selected the largest group of the patent network to perform the link prediction process because a big and complex network is relevant to apply the link prediction methods. Thus, although the number of collected patents is 1,188, the number of all nodes is 1445 by adding the cited patents that are not included in the UAV technology and excluding small groups of patents. Although several patents have many links with other patents, most patents have few links, creating central patents and peripheral patents. Thus, in this patent network, many patents can have new potential links with other patents.

Vector transformation of documents by means of Doc2vec and SAO structures

When the Doc2vec is applied to textual data, to yield more accurate results, the parts of texts that are to be analyzed should be previously identified. We extracted the texts in the abstract part of patent documents. Since the abstract briefly explains the main ideas of patents, the proposed approach uses the abstract part to achieve accuracy and efficiency in text analysis. Thus, we built a database that has bibliometric information, such as patent number, application year, and abstract data of patent documents. We used Python programming to derive the vectors of each document by means of the Doc2vec algorithm. Figure 5 and Table 1 partially present the vectors of patent documents that are derived by the Doc2vec. While we did the Doc2vec learning, the patent's SAOs and documents were embedded into the 50-dimension vector spaces with 55 iterations of learning, and only the document vectors were extracted for analysis.

Similarity analysis between documents

The similarity between documents should be calculated by using the document vectors, in order to predict the potential links between documents. Figure 6 and Table 2 show a symmetric matrix that has the values of cosine similarity between documents. The diagonal cells have a value of 1, because each cell represents its similarity to itself. The cell (d_i, d_j) shows the similarity of the ith document and jth document. For example, the similarity of d_1 and d_2 is 0.10854. The similarity matrix is symmetric, because the similarity has no direction. Thus, the similarities of cells (d_i, d_j) and (d_j, d_i) are the same.

In order to judge which relationships of unconnected patents in a network can be connected, the similarity between patents should be investigated. If a cut-off value is set for the decision on the connection, a relationship that has a higher value than the cut-off value can

	A	B	C	D	E	F	G	H	I	J	K	L
1	*dt_1	1.268011	-0.36759	-0.3472	-0.47052	1.79828	2.091455	-0.06211	-0.19958	-0.62553	0.042465	-0.92822
2	*dt_2	1.277277	0.025093	-0.40978	1.328572	1.672683	0.216598	-1.80478	0.009438	-0.55312	1.416989	-1.31541
3	*dt_3	-0.09495	-1.08873	-0.88355	1.320164	1.170413	0.057899	-0.40227	-0.85075	-0.06981	0.474729	-1.12152
4	*dt_4	1.232299	-0.1019	0.198711	-0.50032	0.568049	0.443045	-1.61689	0.259199	-0.36252	1.306585	-0.79077
5	*dt_5	1.38023	-0.33617	-0.28438	-0.52913	1.750757	2.21389	-0.26841	-0.11533	-0.37018	0.006431	-0.63139
6	*dt_6	0.401451	-1.02097	-0.04921	1.424566	0.538437	0.450917	-0.59509	-0.60008	0.374294	0.713994	-1.06412
7	*dt_7	0.096255	-0.25283	-0.57634	-0.34927	0.648152	0.580625	-1.44627	-0.05807	-0.37677	1.061473	-1.98055
8	*dt_8	0.742562	-1.16738	-0.08983	0.435679	-0.18851	-0.02516	-0.56855	-0.52305	-0.28883	0.794322	-0.53172
9	*dt_9	0.815747	-1.26399	-0.17737	0.435734	-0.0487	-0.16416	-0.72992	-0.41572	-0.33013	0.868314	-0.41606
10	*dt_10	0.859195	-1.24916	-0.1079	0.347744	0.006766	0.106625	-0.59841	-0.65246	-0.09263	0.959401	-0.52891
11	*dt_11	0.6978	-1.23781	-0.10308	0.394368	-0.06385	0.148237	-0.70956	-0.52168	-0.13001	0.802579	-0.4893
12	*dt_12	0.740429	-1.20877	-0.13128	0.334892	-0.12193	0.108683	-0.65518	-0.65063	-0.15359	0.823512	-0.42681
13	*dt_13	0.787677	-1.2131	-0.1251	0.371736	0.008466	0.094733	-0.67581	-0.38345	-0.26256	0.79678	-0.32382
14	*dt_14	0.121536	0.035058	-0.13763	0.2036	-0.00261	0.163664	0.432594	-0.01567	-0.26953	0.642241	-0.5923
15	*dt_15	-0.06872	-0.47581	-0.88697	0.342293	0.800023	0.331492	0.661297	1.36198	-1.19805	1.087275	0.304806
16	*dt_16	0.143826	-1.46471	-0.48594	2.000025	0.112661	0.930608	-1.16195	-0.59185	-0.21821	1.305221	-0.94543
17	*dt_17	0.315412	-0.01768	-0.61047	-0.37366	0.579906	0.469253	-1.65811	-0.15145	-0.32777	1.225598	-2.17508
18	*dt_18	1.012589	-1.3151	-0.23151	0.323534	-0.15421	0.177617	-0.73503	-0.50961	-0.1193	0.883144	-0.38645
19	*dt_19	1.448068	-1.31105	-0.01833	0.030472	-0.05349	-0.52577	-0.24371	-1.3001	1.278742	0.20624	-1.78054
20	*dt_20	0.918673	-0.46995	0.371378	0.162735	-0.0614	0.549599	-1.16805	-0.95308	-1.28942	0.389855	0.176395
21	*dt_21	0.612081	-1.04674	0.085489	1.247887	0.80961	0.364889	-1.85914	0.380046	-0.4436	-0.07234	0.22389
22	*dt_22	0.837897	-0.55726	0.067841	0.363088	-0.28239	0.034976	-0.49523	-0.45574	-0.53342	0.430061	0.496485
23	*dt_23	0.131926	-2.50626	-0.19442	0.725794	-0.69538	0.495887	-0.7638	-0.61652	0.579639	0.710822	-0.71844

Fig. 5 Results of Doc2vec

Table 1 Examples of Doc2vec results

	V1	V2	V3	V4	V5	V6	…
US10000284	0.522	−0.040	−0.151	0.456	1.544	0.011	…
US10001776	1.306	−1.603	0.871	0.197	−0.461	0.124	…
US10001778	1.578	0.548	0.270	1.036	1.460	1.220	…
US10005555	1.250	0.098	−0.363	0.514	1.216	0.961	…
US10005556	2.095	−1.108	−0.748	0.523	–0.250	−0.768	…
US10006747	-1.073	−1.556	−1.701	0.402	−0.518	1.153	…
…	…	…	…	…	…	…	…

be considered as a potential link. Thus, the selection of the cut-off value is critical in the proposed approach that aims at predicting links. In this paper, the cut-off value can be derived by calculating the average similarity of linked patents, because such cases provide reliable criteria for finding potential links. For this, we searched pairs of patents that have citation relationships in an actual citation matrix, and then drew their similarities from the similarity matrix. Figure 7 shows that the citation matrix consists of the actual relationships of citation between patents. In this matrix, a cell that a pair of patents has a citation relationship has 1, whereas if two patents have no citation relationship, the cell should have 0. Since the diagonals of the matrix are cells for a document itself and a patent cannot cite itself, the cells have 0. We identified two groups of relationships between patents (linked relationships and non-linked relationships), to derive the cut-off value from the citation matrix. Consequently, the similarity values of non-linked relationships and linked relationships are derived on average. Whereas the average similarity of the linked group is 0.403184, that of the non-linked group is 0.268788. Since predicting a link in a network needs to be rigorous and conservative, we selected the average similarity of the linked group to set the cut-off value. Thus, if

	A	B	C	D	E	F	G	H	I	J	K	L
1		*dt_1	*dt_2	*dt_3	*dt_4	*dt_5	*dt_6	*dt_7	*dt_8	*dt_9	*dt_10	*dt_11
2	*dt_1	1	0.10854	0.21323	0.029377	0.284572	0.222555	0.154084	0.061389	0.286805	0.103356	0.335523
3	*dt_2	0.10854	1	0.392317	-0.09038	0.227639	0.028818	0.166094	0.051826	0.03283	0.19811	0.245046
4	*dt_3	0.21323	0.392317	1	0.276838	0.334647	0.115861	0.330276	0.270815	0.400264	0.246199	0.338811
5	*dt_4	0.029377	-0.09038	0.276838	1	0.238039	-0.07975	0.24706	0.27801	0.388737	0.090917	0.11704
6	*dt_5	0.284572	0.227639	0.334647	0.238039	1	0.162448	0.260951	0.217912	0.373862	0.006988	0.576262
7	*dt_6	0.222555	0.028818	0.115861	-0.07975	0.162448	1	0.201162	0.194418	0.263107	0.278213	0.273648
8	*dt_7	0.154084	0.166094	0.330276	0.24706	0.260951	0.201162	1	0.017175	0.293726	-0.06951	0.094156
9	*dt_8	0.061389	0.051826	0.270815	0.27801	0.217912	0.194418	0.017175	1	0.188563	0.090038	0.202349
10	*dt_9	0.286805	0.03283	0.400264	0.388737	0.373862	0.263107	0.293726	0.188563	1	0.19105	0.375762
11	*dt_10	0.103356	0.19811	0.246199	0.090917	0.006988	0.278213	-0.06951	0.090038	0.19105	1	0.17594
12	*dt_11	0.335523	0.245046	0.338811	0.11704	0.576262	0.273648	0.094156	0.202349	0.375762	0.17594	1
13	*dt_12	0.227146	0.049837	0.28554	0.108994	0.304936	0.327112	-0.00678	0.378667	0.297964	-0.09678	0.443751
14	*dt_13	0.264034	0.437625	0.570842	0.136499	0.343865	0.075079	0.193609	0.20338	0.314314	0.144106	0.434923
15	*dt_14	0.225888	0.175022	0.195421	0.413439	0.023658	0.111017	0.302775	-0.0103	0.213609	0.402835	0.113887
16	*dt_15	0.116535	0.207049	0.263974	0.112625	0.057564	0.183492	0.304484	-0.00586	0.090048	0.310778	0.019185
17	*dt_16	0.015079	0.161813	0.30822	0.228748	0.375279	0.132775	0.234472	0.298429	0.143369	0.247773	0.357944
18	*dt_17	0.368435	0.042943	0.352578	0.495206	0.288567	-0.03672	0.294963	0.103489	0.530904	0.326305	0.393491
19	*dt_18	0.083518	0.093147	0.328919	0.145233	0.247399	0.207586	0.323711	0.339971	0.214861	0.318736	-0.02124
20	*dt_19	0.179616	0.196875	0.389131	0.004709	0.04762	0.469008	0.136346	0.241432	0.209363	0.082183	-0.0284
21	*dt_20	0.205755	0.11464	0.251315	0.128712	0.183544	0.156341	0.201421	0.13382	0.116752	0.391674	-0.03854
22	*dt_21	0.246679	0.073523	0.280426	0.276143	0.097253	0.308098	0.085489	0.151074	0.459957	0.417181	0.073523

Fig. 6 Cosine similarity between documents

Table 2 Examples of cosine similarity between documents

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	…
Doc1	1	0.109	0.213	0.029	0.285	0.223	…
Doc2	0.109	1	0.392	−0.090	0.228	0.029	…
Doc3	0.213	0.392	1	0.277	0.335	0.116	…
Doc4	0.029	−0.090	0.277	1	0.238	−0.080	…
Doc5	0.285	0.228	0.335	0.238	1	0.162	…
Doc6	0.223	0.029	0.116	−0.080	0.162	1	…
…	…	…	…	…	…	…	…

	US5087969	US5146702	US5150857	US5155684	US5158351	US5160839	US5167550	US5208750	US5218542	US5240207	US5277380
US5087969	0	0	0	0	0	0	0	0	0	0	0
US5146702	0	0	0	0	0	0	0	0	0	0	0
US5150857	0	0	0	0	0	0	0	0	0	0	0
US5155684	0	0	0	0	0	0	0	0	0	0	0
US5158351	0	0	0	0	0	0	0	0	0	0	0
US5160839	0	0	0	0	0	0	0	0	0	0	0
US5167550	0	0	0	0	0	0	0	0	0	0	0
US5208750	0	0	0	0	0	0	0	0	0	0	0
US5218542	1	0	0	0	0	0	0	0	0	0	0
US5240207	0	0	0	0	0	0	0	0	0	0	0
US5277380	0	0	1	0	0	0	0	0	0	0	0
US5295643	0	0	0	0	0	0	0	0	0	0	0
US5317866	0	0	0	0	0	0	0	0	0	0	0
US5377106	0	0	0	0	0	0	0	0	0	0	0
US5384887	0	0	0	0	0	0	0	0	0	0	0
US5386759	0	0	0	0	0	0	0	0	0	0	0
US5417140	0	0	0	0	0	0	0	0	0	0	0
US5503595	0	0	0	0	0	0	0	0	0	0	0
US5521817	0	0	0	0	0	0	0	0	0	1	0
US5568250	0	0	0	0	0	0	0	0	0	0	0
US5569088	0	0	0	0	0	0	0	0	0	0	0
US5575438	0	0	1	0	0	0	0	0	0	0	1

Fig. 7 A partial citation matrix of UAV patents

the similarity of a relationship between patents is over the cut-off value, the relationship can be depicted as a link. Table 3 shows the similarity of each pair of patents in the two groups.

Then, we carried out the link prediction on test data that have the information on citation relationships and that are not used for training. Table 4 shows the representative cases of test data that have high similarity. Many pairs of patents that are unconnected were very similar, based on Doc2vec-based link prediction using SAO structures. For example, although the pair of patents (US7097137 and US9669946) in the top row of the table are the most similar and can be predicted as a potentially connected relationship, they actually have no link. This result is partially caused by the fact that the applicants and examiners who provide the citation information might miss important citation relationships. However, since the proposed approach might not work well for link prediction, we derived the confusion matrices of link prediction on test data. In general, two indices, of precision and recall, can be used to assess the accuracy of forecasting. From Table 5, in order to check the result of the proposed method, we construct the confusion matrix. From it, the precision, which means the fraction of relevant instances between the retrieved instances, is 0.002 (75/(37,222 + 75)); and the recall, which is the fraction of the total amount of relevant instances that were actually retrieved, is 0.806 (75/(75 + 18)). Although the precision is low, the recall shows that the result correctly predicts the actual links. Therefore, the proposed approach can successfully find missing links in a patent network, requiring careful investigation on the predicted links, because it has low precision.

Validation

The proposed approach can be validated by comparing its results with those of existing approaches. Even though there are many existing approaches for link prediction, the Adamic–Adar technique has been used in many studies. Thus, the results of the proposed approach can be compared with those of the Adamic–Adar technique, applying two approaches to the collected data. In addition, in terms of the vectorization of patent documents, we compared its results with that of a traditional text mining technique that uses keyword vectors.

Link prediction based on the Adamic–Adar technique

Based on the patent network shown in Fig. 1, potential links were predicted by applying the Adamic–Adar technique. This index is a measure to predict links in a network by

Table 3 Similarity of pairs of patents based on citation or no citation relationship example

Pairs of patents with citation relationship			Pairs of patents with no citation relationship		
Patent A	Patent B	Similarity	Patent C	Patent D	Similarity
US9669946	US7097137	0.994	US9889933	US9714088	0.230
US4626696	US4458156	0.993	US9889933	US9669926	0.266
US9669946	US8864069	0.992	US9889933	US9663226	0.182
US8517306	US6874729	0.992	US9889933	US9561852	0.490
US9284049	US9221537	0.992	US9889933	US9527587	0.318
US7097137	US6874729	0.992	US9889933	US9434473	0.324
…	…	…	…	…	…
average	0.403		0.403		0.269

Table 4 Examples of pairs of patents that have high similarity

Patent 1	Patent Title	Patent 2	Patent Title	Similarity	Link
US7097137	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.994	X
US8864069	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.994	X
US8600602	Flight technical control management for an unmanned aerial vehicle	US8515609	Flight technical control management for an unmanned aerial vehicle	0.994	X
US9346543	Unmanned aerial vehicle and methods for controlling same	US9540104	Unmanned aerial vehicle and methods for controlling same	0.994	X
US9346544	Unmanned aerial vehicle and methods for controlling same	US9346543	Unmanned aerial vehicle and methods for controlling same	0.993	X
US9233754	Unmanned aerial vehicle and operations thereof	US9221536	Unmanned aerial vehicle and operations thereof	0.993	X
US4458156	Flywheel propulsion system for automotive vehicles or the like	US4626696	Flywheel propulsion system for automotive vehicles or the like	0.993	O
US20160134358A1	Beam forming and pointing in a network of unmanned aerial vehicles (uavs) for broadband access	US9866312	Beam forming and pointing in a network of unmanned aerial vehicles (UAVs) for broadband access	0.993	X
US9897457	Method and system for controlling vehicles and drones	US20180089773A1	Method and system for controlling vehicles and drones	0.993	X
US9346544	Unmanned aerial vehicle and methods for controlling same	US9540104	Unmanned aerial vehicle and methods for controlling same	0.992	X
US8864069	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.992	O
US6874729	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.992	O
US9075415	Unmanned aerial vehicle and methods for controlling same	US20180084316A1	Unmanned aerial vehicle and methods for controlling same	0.992	X
US9221537	Unmanned aerial vehicle and operations thereof	US9284049	Unmanned aerial vehicle and operations thereof	0.992	O
US6874729	Launch and recovery system for unmanned aerial vehicles	US7097137	Launch and recovery system for unmanned aerial vehicles	0.992	O
US9889930	Unmanned aerial vehicle protective frame configuration	US20180099745A1	Unmanned aerial vehicle protective frame configuration	0.992	X

Table 4 (continued)

Patent 1	Patent Title	Patent 2	Patent Title	Similarity	Link
US7097137	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.994	X
US8864069	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.994	X
US8600602	Flight technical control management for an unmanned aerial vehicle	US8515609	Flight technical control management for an unmanned aerial vehicle	0.994	X

Table 5 Confusion matrix made by Doc2vec-based link prediction

Predicted Actual	Positive	Negative	
Positive	True Positive (TP) 75	False Negative (FN) 18	Recall 0.806
Negative	False Positive (FP) 37,222	True Negative (TN) 88,363	Specificity 0.704
	Precision 0.002	Negative Predictive Value 1.000	Accuracy 0.704

calculating the number of shared links between two nodes. If a pair of nodes has many shared neighbors, it has a high possibility for potential connection. In addition, the index is derived by summing the inverse logarithmic degree centrality of neighbors shared by the two nodes, to ensure that common elements with many neighbors are less significant than are elements with few neighbors. Equation 3 presents the Adamic–Adar index.

$$Adamic - Adar\,(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|} \tag{3}$$

where $N(x)$ is the set of nodes adjacent to u.

Table 6 shows the top 20 pairs of patents that have high scores on the Adamic–Adar index. Most cases between the top 20 pairs include two patents that have similar titles. For example, the pair of US9613538 and US9609288 have the same title, but they have no link in terms of patent citation. Although they have similar neighbors in terms of patent citation relationships, they did not cite each other. Mostly, the patent numbers of the cases are close, which means that those patents were applied for or granted in similar times. Thus, they had little opportunity to cite each other. In addition, the results of link prediction based on the Adamic–Adar technique cannot provide the confusion matrix, because after a patent is granted, the citation relationships cannot be added, and the predicted links cannot be tested by future real links. Thus, we can check the contents of only patents that are predicted to have links. Although most pairs have similar titles and contents, several pairs have totally different technological characteristics. For example, although the pair US8256715 and US8260479 have a high Adamic–Adar value, they are not strongly associated with each other. Whereas US8256715 patent deals with devices, systems, and methods for modular payload integration, US8260479 patent aims at developing modular software architecture. For validation, we compared the results of Doc2vec-based link prediction and the Adamic–Adar index. Some pairs that have low Adamic–Adar value but are similar patents were very similar according to the Doc2vec algorithm. In Table 7, many pairs, such as US9346543 and US9540104, and US4458156 and US4626696, that have high similarity by document embedding and are almost the same, have low Adamic–Adar values. In contrast, there are some cases that are actually similar, as shown by both the proposed approach and the Adamic–Adar value. In Table 6, several pairs of patents that have higher Adamic–Adar values (for example, US9613538 and US9609288, and US9494939 and US10001778) also show higher similarity, based on Doc2vec.

Table 6 Examples of pairs of patents recommended by the Adamic–Adar technique

Patent A		Patent B		Adamic–Adar value	Similarity based on Doc2vec
Patent number	Patent Title	Patent number	Patent Title		
US9613538	Unmanned aerial vehicle rooftop inspection system	US9609288	Unmanned aerial vehicle rooftop inspection system	4.681	0.987
US9494939	Velocity control for an unmanned aerial vehicle	US10001778	Velocity control for an unmanned aerial vehicle	4.214	0.969
US9235218	Collision targeting for an unoccupied flying vehicle (UFV)	US9527587	Unoccupied flying vehicle (UFV) coordination	4.123	0.797
US8256715	Devices, systems and methods for modular payload integration for unmanned aerial vehicles	US8260479	Modular software architecture for an unmanned aerial vehicle	3.770	0.645
US8256715	Devices, systems and methods for modular payload integration for unmanned aerial vehicles	US8515609	Flight technical control management for an unmanned aerial vehicle	3.770	0.465
US8256715	Devices, systems and methods for modular payload integration for unmanned aerial vehicles	US8874283	Drone for inspection of enclosed space and method thereof	3.770	0.390
US8515609	Flight technical control management for an unmanned aerial vehicle	US8260479	Modular software architecture for an unmanned aerial vehicle	3.770	0.649
US8515609	Flight technical control management for an unmanned aerial vehicle	US8874283	Drone for inspection of enclosed space and method thereof	3.770	0.273
US8874283	Drone for inspection of enclosed space and method thereof	US8260479	Modular software architecture for an unmanned aerial vehicle	3.256	0.251
US9618940	Unmanned aerial vehicle rooftop inspection system	US9513635	Unmanned aerial vehicle inspection system	3.253	0.810
US9618940	Unmanned aerial vehicle rooftop inspection system	US9609288	Unmanned aerial vehicle rooftop inspection system	3.253	0.972
US9618940	Unmanned aerial vehicle rooftop inspection system	US9613538	Unmanned aerial vehicle rooftop inspection system	3.253	0.982
US9513635	Unmanned aerial vehicle inspection system	US9609288	Unmanned aerial vehicle rooftop inspection system	3.172	0.816
US9513635	Unmanned aerial vehicle inspection system	US9613538	Unmanned aerial vehicle rooftop inspection system	3.172	0.811
US8600602	Flight technical control management for an unmanned aerial vehicle	US8515609	Flight technical control management for an unmanned aerial vehicle	3.171	0.993
US8028952	System for shipboard launch and recovery of unmanned aerial vehicle (UAV) aircraft and method therefor	US6056237	Sonotube compatible unmanned aerial vehicle and system	2.916	0.094
US8028952	System for shipboard launch and recovery of unmanned aerial vehicle (UAV) aircraft and method therefor	US6567044	Miniature, unmanned remotely guided vehicles for locating an object with a beacon	2.916	0.076

Table 6 (continued)

Patent A		Patent B		Adamic–Adar value	Similarity based on Doc2vec
Patent number	Patent Title	Patent number	Patent Title		
US6567044	Miniature, unmanned remotely guided vehicles for locating an object with a beacon	US6056237	Sonotube compatible unmanned aerial vehicle and system	2.916	0
US5581250	Visual collision avoidance system for unmanned aerial vehicles	US5240207	Generic drone control system	2.911	0.123
US5277380	Toroidal fuselage structure for unmanned aerial vehicles having ducted, coaxial, counter-rotating rotors	US5575438	Unmanned VTOL ground surveillance vehicle	4.681	0.407

Table 7 Adamic–Adar values of linked patents predicted based on Doc2vec example

Patent A		Patent B		Similarity based on Doc2vec	Adamic–Adar value
Patent number	Patent Title	Patent number	Patent Title		
US7097137	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.994	2.026
US8864069	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.994	1.342
US8600602	Flight technical control management for an unmanned aerial vehicle	US8515609	Flight technical control management for an unmanned aerial vehicle	0.993	3.171
US9346543	Unmanned aerial vehicle and methods for controlling same	US9540104	Unmanned aerial vehicle and methods for controlling same	0.993	0.621
US9346544	Unmanned aerial vehicle and methods for controlling same	US9346543	Unmanned aerial vehicle and methods for controlling same	0.993	0.621
US9233754	Unmanned aerial vehicle and operations thereof	US9221536	Unmanned aerial vehicle and operations thereof	0.993	1.857
US4458156	Flywheel propulsion system for automotive vehicles or the like	US4626696	Flywheel propulsion system for automotive vehicles or the like	0.993	0.910
US20160134358A1	Beam forming and pointing in a network of unmanned aerial vehicles (uavs) for broadband access	US9866312	Beam forming and pointing in a network of unmanned aerial vehicles (UAVs) for broadband access	0.993	1.956
US9897457	Method and system for controlling vehicles and drones	US20180089773A1	Method and system for controlling vehicles and drones	0.992	1.833
US9346544	Unmanned aerial vehicle and methods for controlling same	US9540104	Unmanned aerial vehicle and methods for controlling same	0.992	0.621
US8864069	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.992	0.558
US6874729	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.992	0.455
US9075415	Unmanned aerial vehicle and methods for controlling same	US20180084316A1	Unmanned aerial vehicle and methods for controlling same	0.992	0.390
US9221537	Unmanned aerial vehicle and operations thereof	US9284049	Unmanned aerial vehicle and operations thereof	0.991	1.135

Table 7 (continued)

Patent A		Patent B		Similarity based on Doc2vec	Adamic–Adar value
Patent number	Patent Title	Patent number	Patent Title		
US6874729	Launch and recovery system for unmanned aerial vehicles	US7097137	Launch and recovery system for unmanned aerial vehicles	0.991	0.558
US9889930	Unmanned aerial vehicle protective frame configuration	US20180099745A1	Unmanned aerial vehicle protective frame configuration	0.991	0.621
US7097137	Launch and recovery system for unmanned aerial vehicles	US9669946	Launch and recovery system for unmanned aerial vehicles	0.994	0.455
US8864069	Launch and recovery system for unmanned aerial vehicles	US8517306	Launch and recovery system for unmanned aerial vehicles	0.994	1.342
US8600602	Flight technical control management for an unmanned aerial vehicle	US8515609	Flight technical control management for an unmanned aerial vehicle	0.993	0.721

Link prediction based on TF-IDF

Word vectors can be used to vectorize patent documents for link prediction. For this, we chose 20 words that represent all words extracted from collected documents by using TF-IDF, and constructed word vectors by the frequency of each word. Then, the document vectors could be derived by applying the Doc2vec process. We calculated the similarity between documents by using cosine similarity based on document vectors. In order to generate links between patent documents, the cut-off value should be set. We calculated the averages of similarity between connected patents, as well as non-connected patents. Since the similarity values of connected links and non-connected links are 0.453665 and 0.410054 on average, respectively, the average of two values becomes 0.4318595. Since two groups can be divided by the average value, the cut-off value can be 0.4318595. Thus, if the similarity between patent documents is over the cut-off value, the documents can be linked. Given this criterion, we predicted all links between patents and constructed the confusion matrix shown in Table 8. The precision is 0.001 (=368/(258,935+368)), whereas the recall is 0.640 (=368/(368+207)). As this paper presented in "Similarity analysis between documents" section, the precision and recall of the proposed approach are 0.002 and 0.806 respectively. With this result, we can conclude that the proposed method is more precise than are other link prediction methods (the word-vector method). A specific description will be displayed in "Results" section. Table 9 shows that none of the pairs of patents that are most similar based on the word vector technique had any link.

Discussion

In this paper, we suggested a new approach to predicting potential links that have not been connected in a network, using the Doc2vec algorithm and SAO structures. In order to demonstrate the applicability of the proposed approach, we investigated the results of link prediction from several perspectives. First, we generated a confusion matrix, which is generally used in a prediction problem for validation to calculate important indices, such as precision and recall. We found that the precision value was relatively low, whereas the recall value was very high. In principle, the results of prediction based on text mining can hardly be evaluated with only such indices obtained from a confusion matrix. Although natural language might contain subtle and fuzzy meanings, currently text mining cannot catch delicate differences. Thus, in unstructured data analysis, the results of validation are relatively worse than in structured data analysis, because textual data, a representative unstructured data, have many variables, that is, keywords. Sometimes, unstructured data

Table 8 Confusion matrix by word vector approach

Predicted Actual	Positive	Negative	
Positive	True Positive (TP) 368	False Negative (FN) 207	Recall 0.640
Negative	False Positive (FP) 258,935	True Negative (TN) 413,269	Specificity 0.625
	Precision 0.001	Negative Predictive Value 1.000	Accuracy 0.625

Table 9 Examples of pairs of patents recommended by the word vector technique

Patent A		Patent B		Similarity	Link
Patent number	Patent Title	Patent number	Patent Title		
US5786545	Unmanned undersea vehicle with keel-mounted payload deployment system	US5690041	Unmanned undersea vehicle system for weapon deployment	0.989	X
US9643722	Drone device security system	US20180157259A1	Drone device for monitoring animals and vegetation	0.986	X
US9688402	Systems, methods, and devices for unmanned aerial vehicle dispatch and recovery	US9514653	Systems, methods, and devices for providing assistance to an unmanned aerial vehicle	0.974	X
US9569972	Unmanned aerial vehicle identity and capability verification	US9542850	Secure communications with unmanned aerial vehicles	0.962	X
US9911346	Unmanned vehicle, system and method for correcting a trajectory of an unmanned vehicle	US10025315	Unmanned or optionally manned vehicle, system and methods for determining positional information of unmanned or optionally manned vehicles	0.952	X
US8930044	Multi-part navigation process by an unmanned aerial vehicle for navigating to a medical situation	US9436181	Multi-part navigation process by an unmanned aerial vehicle for navigation	0.944	X
US8930044	Multi-part navigation process by an unmanned aerial vehicle for navigating to a medical situation	US9823654	Multi-part navigation process by an unmanned aerial vehicle for navigation	0.944	X
US9811084	Identifying unmanned aerial vehicles for mission performance	US9671790	Scheduling of unmanned aerial vehicles for mission performance	0.935	X
US9583006	Identifying unmanned aerial vehicles for mission performance	US9671790	Scheduling of unmanned aerial vehicles for mission performance	0.935	X
US9881022	Selection of networks for communicating with unmanned aerial vehicles	US9881021	Utilization of third party networks and third party unmanned aerial vehicle platforms	0.933	X

analysis can work well in supervised learning problems, such as spam e-mail classification. However, since the prediction problem is very complex, prediction based on text mining normally does not work well. In particular, the link prediction problems normally have an imbalance issue because there are many nodes in a network but a node has a small number of links with other nodes. Thus, in this type of problem, we need to aim at finding many 'true positive' cases to help analyzers plan their actions. In this paper, our results show a high recall value in link prediction, meaning that the proposed approach can find many cases that can be connected in the future. In terms of precision, although the precision of the proposed approach is low, it is relatively higher than that of word vector technique. If we consider the use of text mining techniques in link prediction, we can apply the proposed approach rather than the word vector technique.

Second, compared to existing link prediction and text mining approaches, our proposed method has several advantages. First of all, the word vector that has been frequently used in text analysis does not work as well as does the Doc2vec-based approach considering SAO structures in link prediction. As we can see on the confusion matrix in Tables 5 and 8, the precision is hugely different between the word-vector and proposed methods. Since the word vector cannot consider the context where a word is used in a specific document, it generally does not provide accurate text analysis. Thus, the proposed approach can present better results of link prediction by reflecting the contextual relationships between SAOs that appear in patent documents. In addition, when we compare the Doc2vec-based approach and the Adamic–Adar technique, we can claim that the former can outperform the latter. In the Adamic–Adar technique, many pairs of unconnected patents that had similar or identical titles mostly had high index values. Although such patents are applied for at patent offices at a similar time, they have little opportunity to cite each other. However, the proposed approach could find such relationships as well, because the similarities of pairs that were predicted by the Adamic–Adar index were high based on Doc2vec. Most importantly, a sophisticated link prediction approach needs to find plausible links that can potentially be connected, rather than links that can obviously be connected. Even though the Adamic–Adar index can retrieve future links in a cohesive area, it cannot explore impactful links in weak relationships. In particular, in R&D planning or marketing, unexpected potential links can provide more insights to managers and analyzers, because unserved customers or unrelated R&D fields are highly critical for market expansion or technology fusion. Therefore, when the purpose of link prediction is to explore new opportunities that are not obvious or are in weak relationships between patents, the proposed approach can be more strongly recommended than the traditional approaches can be.

Finally, the Doc2vec-based link prediction approach can be actively applied to patent management, because the amount of patent data is large, and the relationships between patents are reliable and enable a patent network to be clearly created. A patent network can be developed and used to conduct R&D management. Various types of network can be generated, because nodes can be patents in a patent network, and assignees in an expert network. The main application of the proposed approach to patent management is to explore new opportunities for technology fusion. Since a pair of patents that have similar technological contents but no link can be found in a patent network, they can provide a good opportunity for technology fusion. Although other existing link prediction methods can find new links, they may find only obvious future links, which means that in technology opportunity analysis, the process might be meaningless. The potential links derived by such techniques might be interpreted as potential links that assignees or examiners should have cited in the patent document. Thus, the Doc2vec-based link prediction has a strong advantage in exploring valuable potential links from the weak relationships between patents for developing fusion

technology. Along with that, it can be also applied to a collaborative network of researchers or institutions. An inventor or researcher might want to co-work with other experts in order to conduct research collaboration, because it can be more effective and plausible to develop fusion technology and collectively use a broad spectrum of knowledge, rather than stand-alone knowledge or technology. If we explore the potential links between inventors or institutions in an expert network by link prediction, R&D collaboration is possible for open innovation. For this, a collaborative network should be constructed by re-organizing a patent network with the information on the assignees and affiliations of patents. After developing a patent network and predicting links by using the Doc2vec algorithm, the collaborative network can be built by aggregating nodes and links of patent networks, including predicted links. In conclusion, new collaborators that an assignee can co-work with for new technology development can be explored in the network.

Conclusions

This paper proposes a new approach to predicting potential links in a network by using the Doc2vec algorithm. Although there are many link prediction techniques, they are not appropriate for linking predictions for a patent network. Since they are generally applied to a social network, we need a proper approach to analyzing a patent network that is developed by their citation information and has textual information. For this reason, we used SAO structures, which provide ample information that is extracted from patent documents for link prediction. In analyzing patent documents by means of text mining, the selection of a way to generate document vectors is extremely important in order to produce meaningful results. Thus, we chose the Doc2vec algorithm to correctly create document vectors by reflecting the contexts of technological keywords. We compared the results of word vector-based link prediction and Doc2vec-based link prediction, and showed that the proposed approach works better than does using word vectors. In particular, the Doc2vec approach can work better in predicting the link prediction of patent networks than do any of the traditional link prediction techniques, including the Adamic–Adar index, which has some limitations in dealing with citation networks or document networks.

Although we have suggested a new approach to predicting future links by considering the characteristics of patent documents, we need to discuss the limitations of this paper in terms of algorithm and data analysis. First, our proposed approach does not consider the time dimension in link prediction. Although patent documents have application years, and the time dimension can be used to assess the similarity between patents, we concentrated on the contents of documents and citation information. Second, we did not compare the results of various link prediction techniques with those of the proposed approach. In order to validate the performance of the approach, it needs to be compared with many other existing techniques, such as Jaccard's coefficient and Simrank. Finally, since our purpose was to suggest a Doc2vec-based link prediction process in a patent network, we did not focus on how to apply this approach to patent management. However, it is very important to provide its application to practical works in R&D management. Thus, future work should elaborate this paper to improve the quality of related studies. More advanced research can be done to reflect various factors associated with patent information for link prediction in a patent network and provide specific applications to R&D management, such as R&D collaboration and technology fusion. In addition, the results of link prediction need to be compared with other traditional link prediction approaches.

Appendix 1: Code for Doc2vec

```
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
import os
import tempfile
TEMP_FOLDER = tempfile.gettempdir()
print('Folder "{}" will be used to save temporary dictionary and
corpus.'.format(TEMP_FOLDER))
import warnings
warnings.filterwarnings(action='ignore', category=UserWarning, module='gensim')
from gensim import corpora
import gensim
import numpy as np
import pandas as pd
import csv
import collections
import smart_open
import random
from pprint import pprint # pretty-printer
#set directory
os.chdir('C:/data)
#function definition
def read_corpus(fname, tokens_only=False):
  with smart_open.smart_open(fname,encoding='iso-8859-1') as f:
    for i, line in enumerate(f):
      if tokens_only:
        yield gensim.utils.simple_preprocess(line)
      else:
        # For training data, add tags
```

```
            yield gensim.models.doc2vec.TaggedDocument(gensim.utils.simple_preprocess(line), [i])

    #file open_learning data file
    f=open('Doc2vec2.csv','rb')
    train_corpus = list(read_corpus(f))
    train_corpus
    #for check
    #train_corpus[6]
    #build a model
    model = gensim.models.doc2vec.Doc2vec(size=50, min_count=2, iter=55)
    model.build_vocab(train_corpus)
    %time model.train(train_corpus, total_examples=model.corpus_count, epochs=model.iter)
    #save a model and word, sentence vector
    model.save('Doc2vecmodel_190525.model')
    model.save_word2vec_format('alldata_final.csv', doctag_vec=True, word_vec=True, prefix='',
fvocab=None, binary=False)
```

Appendix 2: Searching query for UAV technology

TI=(((Unman* or Aviait* or fly* or sky*) and (object or thing or Vehicle)) or Drone) and (((locat* or rout* or speed* or fast* or altitute or mov* or distanc* or heigh*) and (inform* or sens*)) or ((sens* or feel* or detect* or notic*) and (Data or informat* or chang* or different*)and (control* or filter* or merge* or process*)) or ((hardware or thing* or product or object or drone or UAV) and (design* or mak* or produc* or lay* out* or light* or weight* or manufact*)) or ((data* or inform* or sourc* or report) and (send* or transmit* or receiv* or arriv*)))

Acknowledgements This work was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2017R1D1A1B03036213.

References

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social Networks, 25*(3), 211–230.

Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006). Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security.

Behrouzi, S., Sarmoor, Z. S., Hajsadeghi, K., & Kavousi, K. (2020). Predicting scientific research trends based on link prediction in keyword networks. *Journal of Informetrics, 14*(4), 101079.

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 740–750).

Chen, H., Li, X., & Huang, Z. (2005). Link prediction approach to collaborative filtering. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05) (pp. 141–142).

Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. arXiv preprint http://arxiv.org/abs/arXiv:1507.07998.

Getoor, L. (2003). Link mining: A new data mining challenge. *ACM SIGKDD Explorations Newsletter, 5*(1), 84–89.

Getoor, L., & Diehl, C. P. (2005). Link mining: A survey. *ACM SIGKDD Explorations Newsletter, 7*(2), 3–12.

Goldberg, Y., & Levy, O. (2014). word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint http://arxiv.org/abs/arXiv:1402.3722.

Guo, J., Wang, X., Li, Q., & Zhu, D. (2016). Subject–action–object-based morphology analysis for determining the direction of technological change. *Technological Forecasting and Social Change, 105*, 27–40.

Hopcroft, J., Lou, T., & Tang, J. (2011). Who will follow you back?: Reciprocal relationship prediction. Proceedings of the 20th ACM international conference on Information and knowledge management, ACM (2011), pp. 1137–1146.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS), 22*(1), 116–142.

Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers,1, 873–882.

Jeong, B., Ko, N., Son, C., & Yoon, J. (2021). Trademark-based framework to uncover business diversification opportunities: Application of deep link prediction and competitive intelligence analysis. *Computers in Industry, 124*, 103356.

Kroeger P. R., Analyzing grammar: An introduction. Cambridge University Press, 2005.

Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint http://arxiv.org/abs/arXiv:1607.05368.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188–1196).

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Advances in neural information processing systems (pp. 2177–2185).

Li, S., Chua, T. S., Zhu, J., & Miao, C. (2016). Generative topic embedding: A continuous representation of documents. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 666–675).

Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology, 58*(7), 1019–1031.

Liu, Y., Liu, Z., Chua, T. S. & Sun, M. (2015). Topical word embeddings. In Twenty-Ninth AAAI Conference on Artificial Intelligence.

Liu, W., & Lü, L. (2010). Link prediction based on local random walk. *EPL (europhysics Letters), 89*(5), 58007.

Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica a: Statistical Mechanics and Its Applications, 390*(6), 1150–1170.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: System demonstrations (pp. 55–60).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint http://arxiv.org/abs/arXiv:1301.3781.

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).

Moehrle, M. G., Walter, L., Geritz, A., & Muller, S. (2005). Patent-based inventor profiles as a basis for human resource decisions in research and development. *R&D Management, 35*(5), 513–524.

Pavlov, M., & Ichise, R. (2007). Finding experts by link prediction in co-authorship networks. *FEWS, 290*, 42–55.

Popescul, A., & Ungar, L. H. (2003, August). Statistical relational learning for link prediction. In IJCAI workshop on learning statistical models from relational data (Vol. 2003).

Rajbabu, K., Srinivas, H., & Sudha, S. (2018). Industrial information extraction through multi-phase classification using ontology for unstructured documents. *Computers in Industry, 100*, 137–147.

Rong, X. (2014). word2vec parameter learning explained. arXiv preprint http://arxiv.org/abs/arXiv:1411.2738.

Sun H. L., Ch'ng E., Yong X., Garibaldi J. M., See S., Chen D.-B. (2017). An improved game-theoretic approach to uncover overlapping communities International Journal of Modern Physics C, 28 (9), 1750112.

Tang, J., Wu, S., Sun, J., & Su. H. (2012). Cross-domain collaboration recommendation. Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1285–129.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1555–1565).

Tang, J., Qu, M., & Mei, Q. (2015, August). Pte: Predictive text embedding through large-scale heterogeneous text networks. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1165–1174). ACM.

Taskar, B., Wong, M. F., Abbeel, P., & Koller, D. (2004). Link prediction in relational data. In Advances in neural information processing systems (pp. 659–666).

Toutanova, K., & Manning, C. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceedings of the 2000 Joint SIGDAT Conference EMNLP/VLC (pp. 63–71).

Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 384–394). Association for Computational Linguistics.

Wu, J., Zhang, G., & Ren, Y. (2017). A balanced modularity maximization link prediction model in social networks. *Information Processing & Management, 53*(1), 295–307.

Xie, Q., Zhang, X., Ding, Y., & Song, M. (2020). Monolingual and multilingual topic analysis using LDA and BERT embeddings. *Journal of Informetrics, 14*(3), 101055.

Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H., & Zhang, G. (2018). Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics, 12*(4), 1099–1117.