World Wide Web (2021) 24:1791–1812
https://doi.org/10.1007/s11280-021-00885-4

Patent2Vec: Multi-view representation learning on patent-graphs for patent classification

Lintao Fang[1] · Le Zhang[1] · Han Wu[1] · Tong Xu[1] · Ding Zhou[1] · Enhong Chen[1]

Received: 30 October 2020 / Revised: 20 February 2021 / Accepted: 21 April 2021 /
Published online: 16 June 2021

Abstract
Patent classification has long been treated as a crucial task to support related services. Though large efforts have been made on the automatic patent classification task, those prior arts mainly focus on mining textual information such as titles and abstracts. Meanwhile, few of them pay attention to the meta data, e.g., the inventors and the assignee company, and the potential correlation via the metadata-based graph has been largely ignored. To that end, in this paper, we develop a new paradigm for patent classification task in the perspective of multi-view patent graph analysis and then propose a novel framework called Patent2vec to learn low-dimensional representations of patents for patent classification. Specifically, we first employ the graph representation learning on individual graphs, so that view-specific representations will be learned by capturing the network structure and side information. Then, we propose a view enhancement module to enrich single view representations by exploiting cross-view correlation knowledge. Afterward, we deploy an attention-based multi-view fusion method to get refined representations for each patent and further design a view alignment module to constraint final fused representation in a relational embedding space which can preserve latent relational information. Empirical results demonstrate that our model not only improves the classification accuracy but also improves the interpretability of classifying patents reflected in the multi-source data.

Keywords Patent classification · Multi-view learning · Network embedding

1 Introduction

Patent classification is of great significance to improve the efficiency of large-scale patent management and services [46]. Generally, each patent will be manually classified into multiple categories by patent examiners according to its domains [34]. However, due to the

This article belongs to the Topical Collection: *Special Issue on Explainability in the Web*
Guest Editors: Guandong Xu, Hongzhi Yin, Irwin King, and Lin Li

✉ Tong Xu
tongxu@ustc.edu.cn

Extended author information available on the last page of the article.

rapid growth of patent applications in recent years, traditional solution with laborious and time-consuming manual operations may hardly meet the demands. Therefore, the automatic patent classification tools are urgently required to support related services.

As far as we are concerned, large efforts have been made to deal with this task. Some studies target at mining distinctive features to classify the patents, including the multiple structured *meta* features (e.g. patent inventor, patent assignee and patent citations) and the unstructured information (e.g. title, abstract, claims and citation networks) in patent documents [22, 45]. And the others focus on adapting various techniques to design special classifiers for better classification results, including SVM [44], CNN [22] and Bert [32].

However, most prior arts mainly focus on one specific aspect for patent classification. In other words, few of them pay attention to the multi-view perspective based on the potential correlations among various meta-features, e.g., words, patents, inventors and assignee companies. Indeed, as shown in Figure 1, we realize that patents are not only directly connected via citations, but also indirectly connected via meta-features, which results in the *multi-view meta-data-based graphs*. Along this line, the explainability of the classification process could be even enhanced by integrating and weighing the multiple sources of metadata. Accordingly, patents with mutual connections may share similar representations. Therefore, a new graph-driven paradigm could be designed by integrating and analyzing the multiple graphs for more comprehensive patent representation, and then support the classification task.

Along this line, however, challenges still exist in how to jointly represent and integrate the individual graphs, in which the different contributions from different views could be measured. To tackle these challenges, in this paper, we propose a novel framework called Patent2Vec for the patent classification task based on the multi-view patent representation. To be specific, we first construct three patent graphs with considering the *word, inventor* and *assignee*, which are illustrated in Figure 1. Afterward, we design a three-stage framework to represent the patent in a multi-view way. In detail, to exploit cross-view correlation knowledge, we propose a view enhancement module to enrich single-view representations by capturing the interactions between different views. Then, to eliminate abundant information, we deploy an attention-based multi-view fusion method to get refined representations for each patent. Moreover, to preserve latent relational information, we design a view alignment to model the dependency relations by mapping final refined representation and single view representations to a shared representation space. At last, the model is trained in a joint learning paradigm by simultaneously optimizing classification and alignment loss. The contributions of this paper can be summarized as follows:

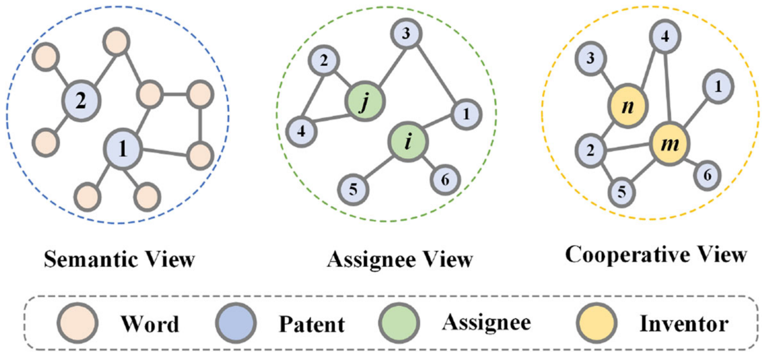

Figure 1 An Example of multi-view graphs

- We propose a novel framework Patent2vec for patent classification in a multi-view graph-based perspective, which is among the first ones to best of our knowledge.
- We design a comprehensive learning approach by capturing cross-view correlation knowledge and modeling dependency between fused representations and single-view representation relations, by which we can obtain effective patent representations.
- Extensive experiments on real-world data proved that patent2vec outperforms several state-of-the-art baseline methods by a large margin, which demonstrates that our model can improve the accuracy and explainability through the construction and analysis of multiple meta-data-graphs.

The rest of the paper is organized as follows. In Section 2, we introduce related works, including patent classification, graph representation learning and multi-view representation learning. In Section 3, we describe the real-world data and formulate the problem. The technical details of our proposed patent2vec framework are presented in Section 4. We then show experimental results and discussions in Section 5. Finally, we conclude our work in Section 6.

2 Related work

In this section, we will summarize the related works as the following three categories, namely patent classification, network embedding and multi-view representation learning respectively.

2.1 Patent classification

Many efforts have been made for automatic patent classification, existing methods can be roughly divided into two aspects: factor mining and specially designed patent classifiers. On the one hand, since there are multiple structured meta features and unstructured information like content and corrections in patent documents, factor mining methods [13, 19, 45] focus on mining distinctive features for classification in patents through qualitative and quantitative analysis. Among these features, textual information [13, 22, 45] is often the most widely exploited factor for distinguishing different types of patents. On the other hand, specially designed patent classifiers [8] involve traditional non-deep learning methods like association rule [11], and support vector machine [25, 44], and deep learning ones like CNN [12, 22, 25], GRU [31], Bert [32], graph neural networks [35] and so on. For example, DeepPatent [22] built a deep convolutional neural network model combined with the word embedding to classify patent documents and PatentBert [32] exploited the powerful pre-training language model Bert [3] and then fine-tuning it to handle multi-label patent classification problem. [22] present an attention-based GCN model over textual graph to solve patent classification problem. However, few of these previous deep learning related studies have focused on this task from a multi-view perspective. Furthermore, existing methods lack explainability, which requires that an intelligent system provides explanations about why patents are classified into a certain category.

2.2 Network embedding

Network embedding aims to embed nodes into a low dimensional space while preserving the network structure and property. The traditional techniques are based on matrix factorization, like LLE [33] and GraRep [1]. Inspired by the skip-gram model [20], some random-walk based models are proposed to deploy truncated random walks on graph for presentation learning, such as Deepwalk [29], Node2vec [9] and metapath2vec [5]. The main difference of these models is the sampling strategy of random-walk, which generates the node sequence similar to word sequence in natural language processing. For example, Deepwalk [29] applies the depth first search strategy to create node sequence, and Node2vec [9] extends Deepwalk by taking advantage of the breadth first search strategy, while they can only be used for the homogeneous graph. And metapath2vec [5] proposes a new strategy to consider the heterogeneity of graph. TFE [47] employs random walks on both original and transpose networks to learn representations for competitive analysis. Recently, some graph neural networks are proposed to capture the rich neighborhood information to represent nodes. For example, GCN [17] performs graph convolution by aggregating neighborhood information to learn embedding. GraphSAGE [10] iteratively generates the node embedding by aggregating features from its sampling neighborhood. GAT [37] applies the attention mechanism for aggregating representation of neighbors to parallel update node representation. SOPE [4] integrates heterogeneous information and multiple relations to generate representations in signed network.

2.3 Multi-view representation learning

Representation learning aims at learning low-dimensional vector of the data and then applies it to downstream tasks [2, 39, 41, 43]. However, multi-view representation learning targets at exploring how to utilize multi-view data for representation learning. The existing techniques can be roughly divided into two categories. The first one is multi-view representation fusion, which aims to integrate multi-view data into a single representation to comprehensively represent data [24]. Representative methods include graphical model-based fusion and neural network based fusion, consisting of multi-view autoencoder [48], multi-view convolutional neural network [7] and multi-view recurrent neural network [16]. For example, [48] built an autoencoder to fuse multiple views into a single refined representation, which is then utilized to perform link prediction. [40, 42] introduce a collective fusing strategy to fuse representations from multi-view transition graphs to detect risky areas. The other one is called alignment, which tries to capture relations among multiple different views. For example, canonical correlation analysis based method DeepCCA [38] and CM-GANs [28] that use GANs to model cross-view joint distribution. [27] employs deep CNNs to align multi-view representations.

3 Preliminaries

In this section, we first introduce the real-world dataset in our study and then discuss the insights we found in the data. Finally, we formulate the problem of multi-view patent representation task for patent classification.

3.1 Pre-study of dataset

The data utilized in this paper are collected from USPTO (the United States Patent and Trademark Office), one of the largest patent granting organization in the world. The dataset contains more than 7 million patents belonging to about 1 million companies and more than 3 million inventors. Detailed statistics of the dataset are shown in Table 1. The patent classification is different from traditional text classification problems due to the inherent special

Table 1 The statistics of the dataset

Statistics	Values
The number of patents	614 943
The number of classes	443
The number of inventors	82 977
The number of companies	6 859
The number of words	126 988

characteristics of patent data, such as relations between patents and inventors, patents and companies, which might have an extensive effect on promoting patent classification. To reveal this point, we select some representative companies and inventors and measure the distributions of different patent categories.

Figure 2 illustrates patent categories distribution of selected companies and inventors, from which we can find that patents' categories are highly related to its corresponding companies and inventors. Besides, every company or inventor holds a latent unique distribution over patents categories. Take Microsoft as an example, more than 40% of its patents belong to the G06F category, which refers to "`ELECTRIC DIGITAL DATA PROCESSING`". Further statistics reveal that over 80% of patents for each company nest in only a small set of about 5% categories and over 90% of patents for each inventor nest in about 10 categories. With this clue, we can incorporate proper representations of companies and inventors to boost the patent representation for achieving more accurate classification results.

3.2 Formulation of patent graphs

With the patent records and above insights, we construct multi-view patent graphs to formally describe the dataset, which is defined as follows:

Definition 1.1 (Semantic View Graph) The semantic view graph is defined as $G^s = (V^p \cup V^w\ ,\ E_{ww} \cup E_{pw})$, where V^p denotes the set of patent, V^w denotes the set of word, E_{ww} denotes the word-word edge set and E_{pw} denotes the patent-word edge set. Each edge in E_{ww} indicates the positive Point-wise Mutual Information (PMI) score between two words. The PMI score is calculated by $PMI\left(w_i, w_j\right) = \log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$, where $p(w_i, w_j)$ represents the frequency of co-occurrence of two words w_i, w_j in a sliding window, and

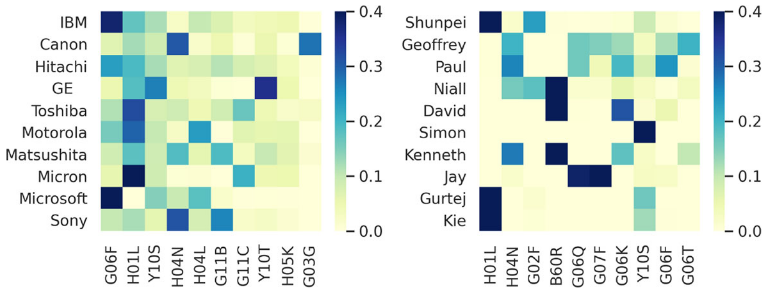

Figure 2 Heatmap of patent category distributions w.r.t. companies and inventors. The horizontal axis denotes the patent category and the vertical axis denotes the company and inventor respectively. The darker the color, the greater the number

$p(w_i)$ denotes the word frequency. Each edge in E_{pw} indicates whether a patent contains a word, which is undirected and unweighted.

Definition 1.2 (Assignee View Graph) The assignee view graph is defined as $G^a = (V^p \cup V^c\ ,\ E_{pp} \cup E_{pc})$, where V^p denotes the set of patent, V^c denotes the set of company, E_{pp} denotes the patent-patent edge set and E_{pc} denotes the patent-company edge set. Each edge in E_{pp} indicates the citation relationship between two patents and each edge in E_{pc} indicates whether the patent belongs to the company.

Definition 1.3 (Cooperative View Graph) The cooperative view graph is defined as $G^c = (V^p \cup V^i\ ,\ E_{pp} \cup E_{pi})$, where V^p denotes the set of patent, V^i denotes the set of inventor, E_{pp} denotes the patent-patent edge set and E_{pc} denotes the patent-company edge set. Each edge in E_{pp} indicates the citation relationship between two patents and each edge in E_{pc} indicates whether the patent is proposed by the inventor.

3.3 Problem formulation

Following the above definitions, we present the problem formulation of multi-view patent representation for patent classification. In general, we target at classifying patents into multiple categories. Obviously, the different patent graphs we build can describe patents from different perspectives, therefore we can integrate information from the multi-view graphs to make better classifications. Inspired by network representation learning, we formulate the patent classification as the task of node classification over the multi-view patent graphs, and then to enable the node classification task, we push forward representation learning over the patent graphs to learn unified and optimal representations for patents. Formally, the studied problem can be defined as the task of multi-view patent representation for patent classification:

Definition 2.1 Given the multi-view patent graphs $\mathcal{G} = \{G^s, G^a, G^c\}$, which represents the semantic, assignee and cooperative view graph respectively. For each patent $p_i \in V^p$, we aim to learn the potential low-dimensional vector representation $P_i \in \mathbb{R}^d\ \ (d \ll |V^p|)$. The generated representation is then utilized to handle patent classification problem.

4 Patent2vec framework

In this section, we first present a general description of our Patent2vec framework. Then we introduce each component of the framework in detail and finally illustrate the optimization objective.

4.1 Overview

In this paper, we propose a novel framework called Patent2vec to learn low-dimensional representations of patents for patent classification task from the perspective of multi-view graph analysis. Specifically, we first employ the graph representation learning on individual graphs to capture the network structures and side information. Then, we design an enhancement module to fully boost the representations from a single view by incorporating information from other views. After that, the individual representations from single views are fused as a unified compact representation through the attention mechanism.

Furthermore, to make sure the unified representation preserves the information from single view, we design a cross-view alignment constrain to tackle the inductive bias and disagreement between the fusion embedding with the corresponding embedding from single view. Finally, the fusion representations of patents are feed to a fully connected layer to classify patents into multiple categories. By jointly optimizing classification loss and alignment loss, we can obtain patent representations and perform classification task simultaneously. Figure 3 shows the process of incorporating multi-view information to embed patent in a unified representation for patent classification. In the following parts, we will elaborate on the technical details of each component.

4.2 View-specific representation

In this part, we will introduce the representation learning on three graphs, which represent the semantic view, assignee view and cooperative view respectively.

4.2.1 Semantic view

Semantic information plays a fundamental role when classifying patent documents into certain categories. For each patent document, we utilize its title and abstract as the semantic information sources. Inspired by the recent achievement of patent semantic learning [35], we do not distinguish between the types of patent and word nodes and apply GraphSAGE [10] model to learn node embedding to improve operating efficiency and representation performance.

At first, GraphSAGE samples neighbors $N(v)$ for each node v to improve efficiency, then it recursively updates embedding for each node by **Aggregating** and **Updating** operations. In the aggregating stage, node v aggregates the information from its immediate neighbors $N(v)$ by the aggregator functions AGG_k, where k denotes the search depth. Here we choose the long-short-memory network (LSTM) as aggregator for its expressive ability. Given a sequence of neighbors embedding, the aggregating process is defined as:

$$\mathbf{h}^k_{N(v)} \leftarrow \mathrm{AGG}_k\left(\mathbf{h}^{k-1}_u, \forall u \in N(v)\right), \tag{1}$$

where $\mathbf{h}^k_u$ denotes the representation of node v in search depth k. In the updating stage, node v's representation $\mathbf{h}^k_v$ is updated by concatenating the aggregated neighborhood vector $\mathbf{h}^k_{N(v)}$

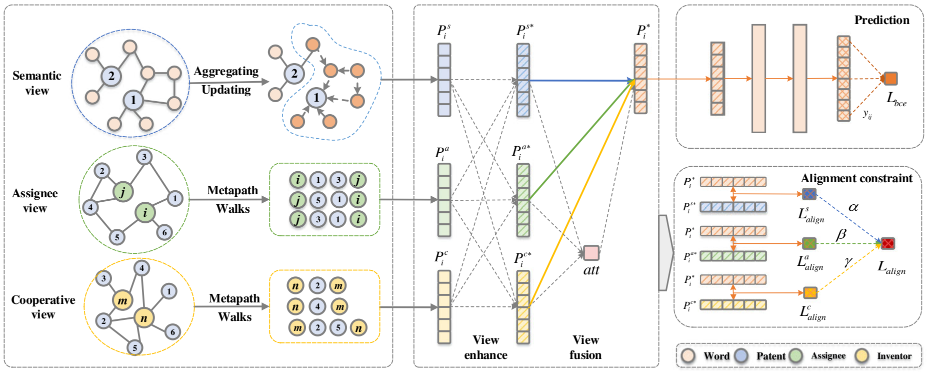

Figure 3 Overview of Patent2vec framework

and its own node hidden state $\mathbf{h}_v^{k-1}$:

$$\mathbf{h}_v^k \leftarrow \sigma\left(W_s^k \cdot concat\left(\mathbf{h}_v^{k-1}, \mathbf{h}_{N(v)}^k\right)\right), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function, and W_s^k is the trainable parameter in depth k. At last, we employ the unsupervised loss to make optimization according to [10], which makes adjacent nodes have similar representations, while the representations of nodes far away from each other have a large difference.

4.2.2 Assignee view

The assignee view graph contains two kinds of nodes, i.e., patents and companies. Different from the above semantic view graph, we consider assignee view graph as a heterogeneous graph since it has a more complex structure. Here, we incorporate the meta-path based random walk to learn node embeddings which is widely adopted in practice to exploit structural properties of a heterogeneous graph. First, we define a meta-path pattern $\mathcal{P} : c_i \xrightarrow{assign} p_j \xrightarrow{cite} p_k \xrightarrow{assign} c_l$, where c_i denotes company i and p_j denotes patent j. This meta-path pattern describes the correlation between c_i and c_l via a path between p_j and p_k. Next, we generate meta paths based on the defined pattern, where each path is a node sequence. Finally, we utilize the skip-gram model proposed in Word2vec [26] to learn representation of each node. The objective function is shown as follows:

$$\sum_{v \in V^{c \cup p}} \sum_{n \in N_c(v) \cup N_p(v)} \log \frac{\exp(Emb_v \cdot Emb_n)}{\sum_{u \in V^{c \cup p}} \exp(Emb_v \cdot Emb_u)}, \tag{3}$$

where $V^{c \cup p}$ denotes the set of company and patent, $N_c(v)$ denotes the company neighbors of node v, $N_p(v)$ denotes the patent neighbors of node v and Emb_v denotes the embedding of v. We can improve efficiency of the objective with the negative sampling method and learn embedding with SGD method [5].

4.2.3 Cooperative view

Similar to the assignee graph, we also employ the meta-path random walk based network embedding method for the cooperative graph. However, the difference lies in that here we define two kinds of meta-path patterns. The first one is $\mathcal{P}_1 : i_m \xrightarrow{invent} p_i \xrightarrow{cite} p_j \xrightarrow{invent} i_n$, which constructs the correlation between inventor m and n via a path between patent i and j. The other one is $\mathcal{P}_2 : i_m \xrightarrow{invent} p_j \xrightarrow{invent} i_n$, which indicates the co-author relationship between two inventors. By executing the random walks on these meta paths and optimizing the following objective, we can obtain the representation for each node on cooperative graph.

$$\sum_{v \in V^{i \cup p}} \sum_{n \in N_i(v) \cup N_p(v)} \log \frac{\exp(Emb_v \cdot Emb_n)}{\sum_{u \in V^{i \cup p}} \exp(Emb_u \cdot Emb_v)}, \tag{4}$$

where $V^{i \cup p}$ denotes the set of inventor and patent, $N_i(v)$ denotes the inventor neighbors of node v, $N_p(v)$ denotes the patent neighbors of node v and Emb_v denotes the embedding of v. The optimize process is similar to the assignee view.

4.3 Single view representation enhancement

Although the graphs from different view can provide unique information, there still exist implicit information and sophisticated correlation knowledge across different views. To obtain better single-view representation, we design a view enhancement module to boost single view representation by aggregating other views representations. As mentioned before, there are three patent graphs $\mathcal{G} = \{G^s, G^a, G^c\}$, which represent the semantic, assignee and cooperative view graph respectively. By employing the graph representation learning methods introduced above on individual graphs, we can obtain the corresponding representation of node for each view. For simplicity, we denote P_i^s, P_i^a and P_i^c as the representation of patent p_i in semantic, assignee and cooperative view. The target of this module is to get the enhanced representations for each single view, namely P_i^{s*}, P_i^{a*} and P_i^{c*} respectively.

Figure 4 shows the graphical representation of single view representation enhancement module. At first, we stack the three single view representations to get the input of the enhancement module $P_i = [P_i^s, P_i^a, P_i^c]$, where $P_i \in \mathbb{R}^{k \times d}$, d is the representation size and k is the number of view, i.e., k=3. It is intuitive that the representations of each view can make different contributions on other views, so that we then leverage the covariance matrix to capture the view-wise correlations among representations from different views.

To be specific, we employ a function $f(\cdot)$ to reduce the size d to $\frac{d}{r}$ of the input tensor, where $f(\cdot)$ consists of a linear transformation followed by a batch normalization layer and a Leaky Rectified Linear Unit (LeakyReLU), r denotes the reduction factor. In addition, we use another transformation to form $g(\cdot)$, which acts like the function $f(\cdot)$. The shapes of $f(P_i)$ and $g(P_i)$ are both $k \times \frac{d}{r}$. After that, following the practice in [23], we design the covariance matrix as follows:

$$\mathbf{\Sigma} = f(P_i) \cdot \bar{\mathbf{I}} \cdot f(P_i)^T, \tag{5}$$

where $\bar{\mathbf{I}} = \frac{1}{d/r}\left(\mathbf{I} - \frac{1}{d/r}\mathbf{1}\right)$, $\mathbf{I} \in \mathbb{R}^{\frac{d}{r} \times \frac{d}{r}}$ denotes identity matrix and $\mathbf{1} \in \mathbb{R}^{\frac{d}{r} \times \frac{d}{r}}$ denotes all-one matrix. Next, we design a self-attention with Σ to model the view-wise correlations. Similar to [6], we use $\frac{1}{\sqrt{d/r}}$ as the scaling factor for the covariance matrix before applying softmax, which yields

$$U_i = softmax\left(\frac{\Sigma}{\sqrt{d/r}}\right) g(P_i). \tag{6}$$

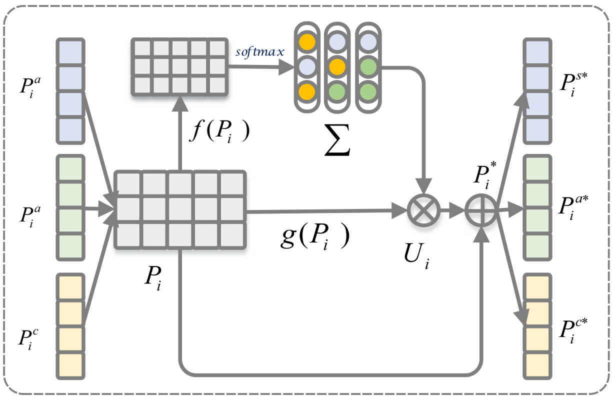

Figure 4 The schematic diagram of representation enhancement module

Finally, we concatenate the original view representation with enrich knowledge to form the new enhanced representations for each view.

$$\boldsymbol{P}_i^* = U_i \oplus P_i \tag{7}$$

where symbol $\oplus$ represents the concatenation operation and $\boldsymbol{P}_i^*$ is a tensor containing the enhanced representations P_i^{s*}, P_i^{a*} and P_i^{c*}.

4.4 Multi-view fusion

The representation of each view provides a specific aspect of information, it is intuitive to integrate them together to get a more comprehensive representation.

Hence, we design an attention-based multi-view fusion module to refine the multiple representations into a single representation, which exploits the complementary information contained in multiple views to comprehensively represent the patent. In this way, we can distinguish the contribution of different views, and also better explain the classification results.

Specifically, the input of fusion module is the enhanced specific view representation, i.e., P_i^{s*}, P_i^{a*}, P_i^{c*}. We utilize the attention mechanism to capture the degree of attention for representing the importance of single view, which is implemented by a two fully connected layer to calculate the attention weights for each view. The fusion module can be formulated as follows:

$$\begin{aligned}
att_m &= W_2^m \cdot \tanh\left(W_1^m \cdot P_i^m + b_1^m\right) + b_2^m, \\
\tilde{att} &= softmax(\tilde{att}), \\
P_i^{m'} &= \tanh\left(W_3^m \cdot P_i^m + b_3^m\right), \\
P_i^* &= \textstyle\sum_m att_m \cdot P_i^{m'},
\end{aligned} \tag{8}$$

where $m \in \{s*, a*, c*\}$ represents the enhanced representation of a specific view, att_m denotes the attention weight of each view m, $\tilde{att}$ is a vector containing att_m ,W_i^m, b_i^m are trainable parameters and P_i^* denotes the final representation of patent p_i.

4.5 Alignment constraint

Though fusion module can compress multi-view embeddings to a single refined representation, however, attention-based fusion may introduce extra noise since different view feature vectors inhabit in different representation spaces. Directly conduct a weighted combination of the multi-view embeddings into a single vector may ignore this issue. Inspired by multi-modal representation learning [18], we propose a view alignment module to address this issue by projecting the single-view representation into a shared representation space and we then further assume that single view embedding of a patent is near to it's fused representation while far from other patent's fused representation in the shared representation space.

To be specific, we first project the fused representation into three view-specific shared representation spaces by the alignment transformation matrices, then we maximize the dot product of the projected fused representation and single view representation to construct the view-specific alignment constraints as follows:

$$\begin{aligned}
\mathcal{L}^{s}_{align} &= \sum_{i} \sigma\left(W_{ps} \cdot P_i^{*}\right)^{T} \sigma\left(W_{ps} \cdot P_i^{s*}\right), \\
\mathcal{L}^{a}_{align} &= \sum_{i} \sigma\left(W_{pa} \cdot P_i^{*}\right)^{T} \sigma\left(W_{pa} \cdot P_i^{a*}\right), \\
\mathcal{L}^{c}_{align} &= \sum_{i} \sigma\left(W_{pc} \cdot P_i^{*}\right)^{T} \sigma\left(W_{pc} \cdot P_i^{c*}\right),
\end{aligned} \tag{9}$$

where W_{ps}, W_{pa} and W_{pc} represent the alignment transformation matrices for three views respectively. After that, we can obtain the final alignment constraint by integrating the view-specific alignment constraints.

$$\mathcal{L}_{align} = \alpha \cdot \mathcal{L}^{s}_{align} + \beta \cdot \mathcal{L}^{a}_{align} + \gamma \cdot \mathcal{L}^{c}_{align}, \tag{10}$$

where α, β, γ denote the weight of constraint.

4.6 Patent classification

Following the above procedure, the fused representations are feed to two consecutive fully connected hidden layers to classifying patents, whose activation functions are sigmoid and ReLU respectively. The final predicted probability $\hat{y}_{ij}$ of p_i for the j-th label is formulated as follows:

$$\hat{y}_{ij} = \text{ReLU}\left(W_2^{T} \cdot \sigma\left({W_1}^{T} \cdot P_i^{*}\right)\right), \tag{11}$$

where W_i^{T} is the trainable parameters. Finally, since we focus on multi-label patent classification task, we apply binary cross entropy (BCE) loss as the objective function as follows:

$$\mathcal{L}_{bce} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{t} \left[y_{ij} \log\left(\hat{y}_{ij}\right) + \left(1 - y_{ij}\right) \log\left(1 - \hat{y}_{ij}\right) \right], \tag{12}$$

where N is the number of training examples and t is the number of categories, y_{ij} is the ground truth label.

Combining the classification loss with alignment constraint, we can get the final objective as follows:

$$\mathcal{L} = \mathcal{L}_{bce} + \lambda \cdot \mathcal{L}_{align}. \tag{13}$$

where λ denotes the balance coefficient of alignment constraint. By jointly optimizing this loss function, we can obtain the representation of each patent and perform classification.

5 Experiments

In this section, we evaluate the performance of our proposed framework Patent2Vec on patent classification task. Specifically, we first introduce the dataset, baselines, evaluation metrics and implementation details. Subsequently, we quantitatively analyze the performance of our Patent2Vec through experiment results comparing, ablation study and parameter sensitivity analysis. Finally, we qualitatively validate the effectiveness of our Patent2Vec by analyzing the contributions of different views and visualizing the learned patent representations.

5.1 Experiment settings

5.1.1 Dataset

Our dataset is retrieved from the USPTO database, which contains about 7 million patents originally. For time efficiency, we sample a subset of the whole database as our dataset. Furthermore, we remove patents that have no inventors nor assignee and eliminate assignee possessing less than 50 patents. In total, we got 614 943 patents, 82 977 inventors, 6 859 assignees and 126 988 words. For notation convenience, we name our dataset as USPTO-600K. We split the data into training and testing set in the ratio of 80% and 20%. Statistics of our dataset is illustrated in Table 2. The classification system we utilized is CPC(Cooperative Patent Classification), which is a three-level hierarchical classification system. The first level, i.e. main class level, has 9 classifications. The second and third levels are subclass level and group level respectively. For example, main class "`G`" represents "`Physics`", subclass "`G06`" represents "`COMPUTING; CALCULATING; COUNTING`" and group "`G06F`" means "`ELECTRIC DIGITAL DATA PROCESSING`". In general, each patent is assigned to several CPC codes.

5.1.2 Evaluation metrics

Patent classification is a multi-label classification task since each patent may belong to multiple categories. Therefore, we employ the rank-based metric including P@k(Precision@K), R@K(Recall@K) and NDCG@K(Normalized Discounted Cumulative Gain), which are widely used in the multi-label classification task. The definitions of these metrics are illustrated as follows:

$$Precision@k = \frac{TP@k}{TP@k + FP@k}, \tag{14}$$

$$Recall@k = \frac{TP@k}{TP@k + FN@k}, \tag{15}$$

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \tag{16}$$

$$NDCG@k = \frac{DCG@k}{IDCG@k}, \tag{17}$$

Table 2 Statistics of multi-view patent graphs

Graph type	Link type	# Links
Semantic-View	patent-word	46 270 156
	word-word	4 404 738
Cooperative-View	patent-inventor	1 199 167
	patent-patent	6 380 514
Assignee-View	patent-company	628 167
	patent-patent	638 0514

where rel_i denotes the relevance of result and IDCG represents the ideal results. In our paper, we set $K = 1, 3, 5$ respectively.

5.1.3 Baselines

In order to demonstrate the performance of our Patent2Vec, we compare it with state-of-the-art methods from three aspects. The first is the representative methods in multi-label text classification field, such as FastXML [30], FastText [15] PfastreXML [14], and recently state-of-the-art methods including DeepPatent [22] and PatentBert [21]. The second is the network representation learning methods, i.e. Deepwalk [29] , Node2vec [9], LINE [36] and GCN [17]. Besides, different from above methods which can only utilize single view of representation, we further use widely used multi-view learning mechanism including Autoencoder and attention-based fusion, which can fuse multi-view representations and then predict patent labels. The details of these baselines are illustrated as follows:

- **FastXML** [30] : FastXML is a tree-based classifiers, are inspired by the ideas of decision tree and build decision trees based on a instances by recursively splitting internal nodes.
- **PFastreXML** [14] : PFastreXML is an extension of FastXML, it prioritizes prediction of tail labels and handles missing labels by proposing the propensity scored loss.
- **BiLSTM**: Here we employ 512 hidden neurons for BiLSTM which takes word embedding as input, and utilize softmax and binary cross entropy loss to make the prediction.
- **FastText** [15] : FastText is an efficient and competitive learning algorithm for word representations and text classification.
- **DeepPatent** [22] : DeepPatent employs convolution and pooling operation on word embedding matrix to extract textual features and then to utilize a fully connected layer to classify documents.
- **PatentBert** [21] : PatentBert applys the pre-trained BERT model(bert-base-uncased) and fine-tune it to perform patent classification.
- **Deepwalk** [29] : Deepwalk takes the truncated random walks for each node to generate the training corpus of node sequences, and then learns the node embedding via maximizing the likelihood of context node prediction from the center nodes.
- **Node2vec** [9] :Different from DeepWalk, it designs a biased truncated random walks to efficiently explore diverse neighborhood and utilize skip-gram model to learn node embedding.
- **LINE** [36] : It learns the node embedding by preserving the first-order proximity or second-order proximity of the network structure separately.
- **GCN** [17] : It obtains node embeddings by aggregating node features on graph. Here we use semantic features learned from word2vec as the input feature of each node, and GCN is conducted on citation graph.
- **AutoEncoder** : We employ a two layer Autoencoder and the the inputs are concatenation of single view representations. We extract hidden layer values as fused representation to make predictions.
- **Attention-Fusion**: Attention-Fusion can calculate single view attention value and then fuse multiple views representations to a refined embedding, which is then utilized to perform classification.

5.1.4 Implementation details

We implement our model based on the PyTorch framework. In the view-specific embedding stage, we set the representation size d as 256 for each view. The number of hidden layers of GraphSAGE is two and the neighbor sample size of each layer is 15 and 10 respectively. As for the assignee view, we set walking length to 40 and walks per node 500. For the cooperative view, we set walking length to 40 and walks per node 50. We set dimension reduction r is to 8 in view enhancement part and fusion dimension as 256 in multi-view fusion stage. The weight of different types of alignment α, β, γ are set to 1.0, 0.2, 0.2 respectively. In the prediction part, the number of hidden units is 1024 and the align loss coefficient is set to 0.01. For model optimization, we set batch size as 128 and the number of epoch 50. We initialized our learning rate as 0.00025 and use an exponential learning rate scheduler to decay the learning rate to 1e-5 until 20 epoch.

5.2 Quantitative analysis

5.2.1 Experimental results

In this part, we first compare our model with the state-of-art methods and illustrate experimental results in Table 3. And we utilize bold-faced to highlight the best experimental results. From Table 3, we can obtain the following observations: First compared with all the baselines, Patent2vec model consistently achieves significant improvements on multiple evaluation metrics. It demonstrates that the patent representations learned by our Patent2vec framework can effectively be utilized to classify patents. Second, we can find that the text-based methods outperform graph-based approaches. It demonstrates that semantic information is more important when predicting patent categories. Third, by comparing single-view and multi-view learning methods, we can observe that multi-view learning can dramatically improve classification accuracy. Furthermore, our method achieves the

Table 3 Overall performance on patent classification

Features	Methods	Precision@K			Recall@K			NDCG@K		
		1	3	5	1	3	5	1	3	5
Text only	FastXML	0.7795	0.4508	0.3102	0.5296	0.7844	0.8567	0.7795	0.7877	0.8117
	PfastreXML	0.7811	0.4514	0.3113	0.5309	0.7857	0.8594	0.7811	0.7889	0.8136
	Bi-LSTM	0.7960	0.4457	0.3074	0.5456	0.7813	0.8526	0.7960	0.7916	0.8151
	FastText	0.8092	0.4574	0.3137	0.5494	0.7962	0.8673	0.8092	0.8056	0.8285
	DeepPatent	0.7890	0.4407	0.3068	0.5361	0.7811	0.8536	0.7890	0.7834	0.8083
	PatentBert	0.8249	0.4535	0.3093	0.5775	0.8009	0.8675	0.8249	0.8178	0.8391
Graph-based	Deepwalk	0.7110	0.3922	0.2724	0.5014	0.7277	0.8064	0.7110	0.7188	0.7495
	Node2vec	0.7204	0.3954	0.2740	0.5082	0.7326	0.8100	0.7204	0.7255	0.7554
	LINE	0.6341	0.3558	0.2501	0.4303	0.6283	0.7035	0.6341	0.6314	0.6583
	GCN	0.7938	0.4443	0.3075	0.5417	0.7804	0.8557	0.7938	0.7880	0.8138
Multiple views	AutoEncoder	0.8261	0.4596	0.3174	0.5625	0.8051	0.8806	0.8261	0.8156	0.8412
	Attention-Fusion	0.8370	0.4673	0.3223	0.5697	0.8150	0.8895	0.8370	0.8269	0.8518
	Patent2Vec	**0.8625**	**0.4871**	**0.3314**	**0.5875**	**0.8367**	**0.9084**	**0.8625**	**0.8516**	**0.8747**

best result among all multi-view methods, this shows that our methods can better utilize multi-view information.

5.2.2 Ablation study

In order to demonstrate the effectiveness of each part in patent2vec, we remove each component of our framework and obtain three variants of our models, namely Patent2vec-Enhance, Patent2vec-Fusion and Patent2vec-Align. For Patent2vec-Enhance, we remove the view representation enhancement part and the input of fusion module is directly the every specific view representation, i.e. P_i^s, P_i^a and P_i^c. For Patent2vec-Fuse, we directly output the enhanced representation as a long vector and perform predictions. Finally, for Patent2vec-Align, we remove the alignment constraint part and train our model only by optimizing the classification loss. The performance results of these variants are illustrated in Table 4.

We can obtain the following conclusions: First, compared with Patent2vec, Patent2vec-Enhance achieves better performance on $P@k$, which proves the effectiveness of our view enhancement part. Second, from the comparison results of Patent2vec and Patent2vec-Fuse, we can illustrate that the proposed attention fusion module can effectively model the weighting relationships of multiple view representations. Third, patent2vec outperform Patent2vec-Align, which can be explained by that the alignment constraint makes the patents with different view aspects closer in the latent vector space. Besides, we can infer the reason for classification according to the weighting distribution obtained by the attention network, which can enhance the interpretability of the embedding learning algorithm. We will illustrate learned weight values and integrate them into the model's interpretable part in the following part.

In order to investigate the effectiveness of each view, we conduct experiments from different types of view combinations, e.g. semantic + assignee and semantic+cooperative. We report results in Figure 5, from which we can notice that combing the assignee and the cooperative view information can boost the performance to a great degree, which is consistent with the assumption in preliminaries. Besides, we can observe that the assignee view and cooperative view hold similar importance. We can achieve about 0.80 P@1 only with semantic view information while about 0.83 with the help of assignee view or cooperative view information. Furthermore, we obtain the best performance when utilizing all of the views. Through the above ablation experiments, we demonstrate the effectiveness of the multi-view learning approach.

5.2.3 Parameter sensitivity

In this section, we study the sensitivity of our model parameters. Specifically, we mainly evaluate how the embedding size d and balance coefficient λ affect the performance. As

Table 4 Performance of variants of Patent2Vec on patent classification

Methods	P@1	P@3	P@5	R@1	R@3	R@5	NDCG@1	NDCG@3	NDCG@5
Patent2vec	0.8625	0.4871	0.3314	0.5875	0.8367	0.9084	0.8625	0.8516	0.8747
Patent2vec-Fuse	0.8573	0.4807	0.3321	0.5842	0.8426	0.9220	0.8573	0.8514	0.8784
Patent2vec-Enhance	0.8438	0.4974	0.3391	0.5569	0.8272	0.8908	0.8438	0.8487	0.8598
Patent2vec-Align	0.8426	0.4713	0.3240	0.5735	0.8202	0.8928	0.8426	0.8326	0.8564

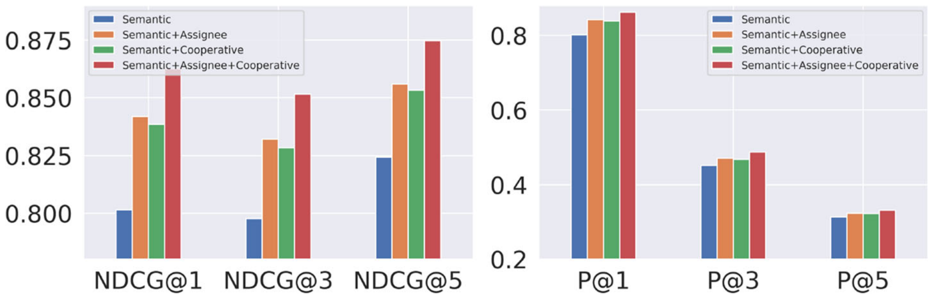

Figure 5 Ablation studies on effect of each views. P@k and NDCG@k results with respect to different types of view combinations

for dimension size d, we set d from 64 to 1024 by multiplication of 2, and present the $P@1$, $R@3$, $NDCG@3$ on patent classification task in Figure 6. It can be observed that the performance increases as the representation size d grows, which could be reasonable as more dimensions can encode more information. However, there is no significant improvement when d is more than 256, which explains why we set d as 256 in our overall experiments to keep the balance between effectiveness and efficiency. In terms of balance coefficient λ, we scale it from 0.0001 to 0.5 and the result is illustrated in Figure 6. As we can see, the performance increases when the coefficient rate climbs. However, the performance decreases when λ continuously increases. The reason is that too large λ may introduce more extra bias and noises which will reduce the performance. Therefore, we set λ as 0.01, which can balance the BCE loss and alignment constraint.

5.3 Qualitative analysis

5.3.1 Analysis of view contribution

In order to investigate each view's contribution when predicting patent's labels, we now focus on the multi-view fusion part. As we previously mentioned, the component aims at fusing multi-view embeddings to a single comprehensive representation. To analyze different views importance, we specify the attention values as view weights, which denote the contributions to the final embeddings. Furthermore, we randomly select some categories

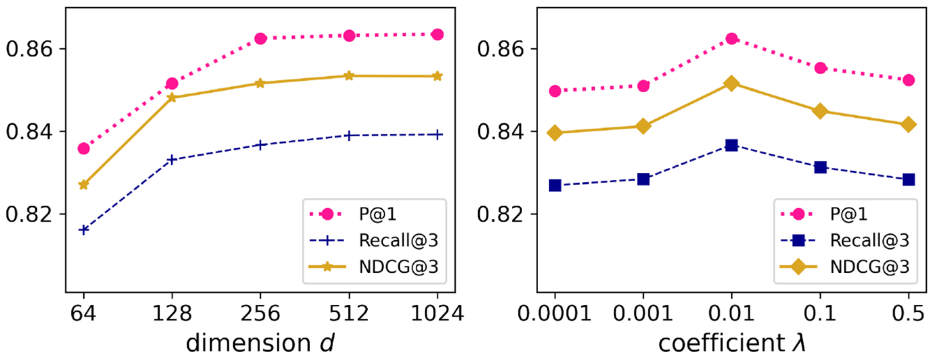

Figure 6 Parameter sensitivity w.r.t the embedding size d and balance coefficient λ

and illustrate the average view weights of each category in Figure 7, from which we can obtain the following observations. First, semantic view accounts for the most contribution of about 40% overall in these categories while cooperative view 30% and assignee view 30%. It demonstrates that semantic features are the most important factor when predicting the classification of patents, which is consistent with observations in ablation studies on the effect of each view in Figure 5. Second, different views play various roles in different categories. In some cases, the assignee view is more important while the cooperative view is more important in other cases, which reflects the different features of various categories. For example, for main class A, cooperative view contributes relatively larger than other classes, since main class A refers to "`HUMAN NECESSITIES`", which is typically more related to the cooperative view than assignee view. Through this analysis, we can observe the contribution of various views more intuitively, so as to better explain our decision basis.

5.3.2 Visualization

Visualization is an important application of network representation learning. Typically, it's done by projecting the original embeddings to a two-dimensional space, so that we can intuitively observe the relevance between nodes in the network. To visualize the generated patent embeddings, we randomly select some patents of several categories and map their

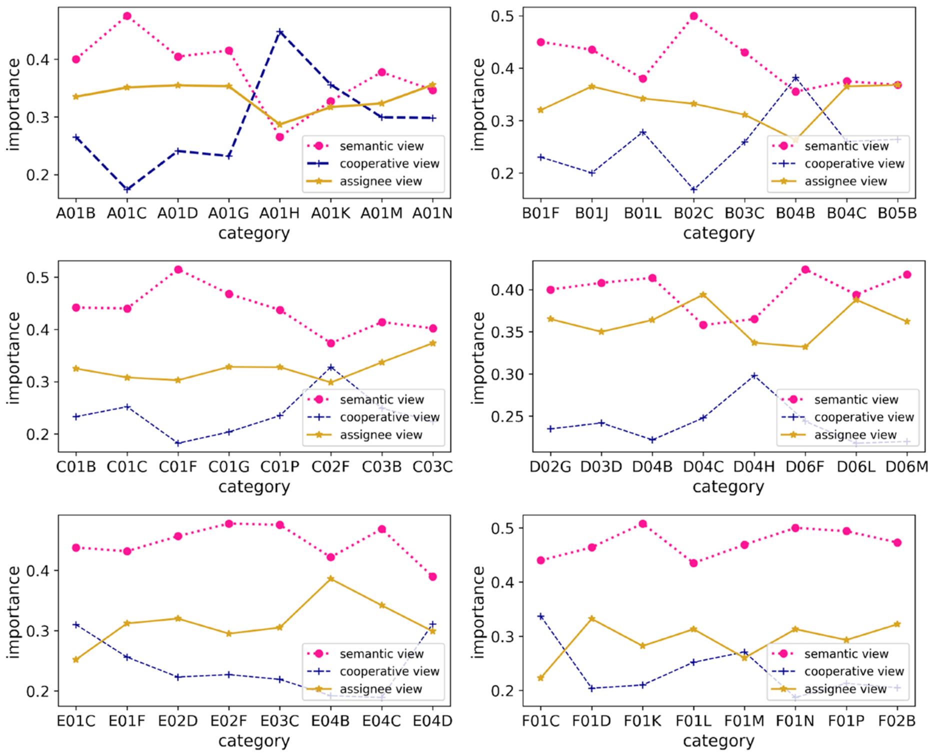

Figure 7 Contribution of different views. The horizontal axis denotes the patent categories from A to H and the vertical axis denotes view importance

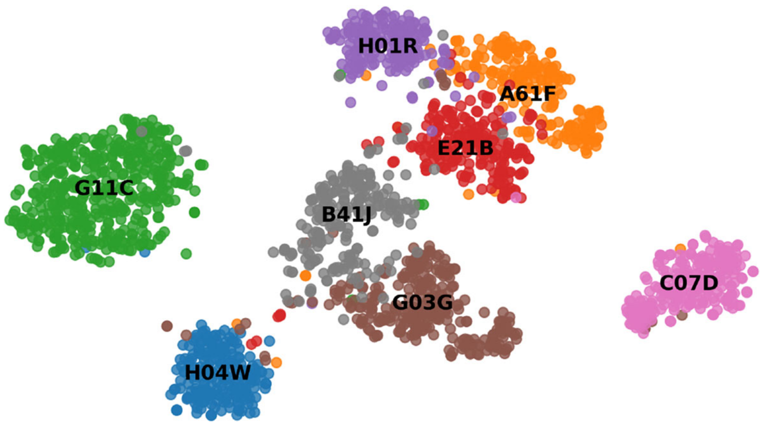

Figure 8 TSNE visualization of patent representations

representations to a two-dimensional space by t-SNE [49]. We present patent embeddings of each category in different colors and annotate the corresponding label at the center of each category's node embedding. From Figure 8, we can observe that there is an obvious distance between different categories' embeddings. It demonstrates that the patent representations learned by our model are discriminative so that they can represent different types of patents in a variety of categories. Meanwhile, there is a small coverage between category embeddings. This result can illustrate there is a certain correlation between category embeddings.

To visualize patent embedding of various companies, we further select several companies from different fields, including information technology, automobile industry, oil industry, and photography industry. Similar to the former one, we also present patents embeddings of each company in different colors and annotate the corresponding company label at the center of the company patents embedding. From Figure 9, we can observe that there is an obvious distance between different field's patent embeddings. Besides, there is a big overlap between General Motors and Toyota. Both companies are in the automobile industry and there exist a lot of patents in common categories. Through Figures 8 and 9, we can further

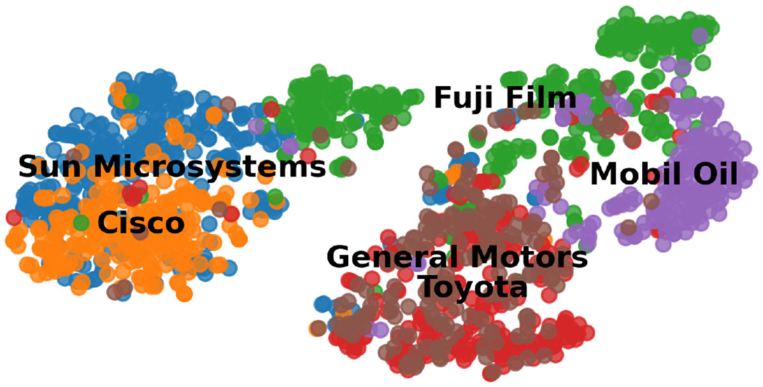

Figure 9 TSNE visualization of company representations

validate the effectiveness of our proposed multi-view representation learning framework and explain the internal mechanism of classifying patents.

6 Conclusion

In this paper, we proposed a novel framework named patent2vec for patent classification in a multi-view network embedding perspective. Considering the unique characteristics of patent data, we built multi-view patent graphs and further designed a sophisticated multi-view enhancement module, which could enrich single view representation by capturing the interactions between different views. Furthermore, we fused multi-view representations and designed an alignment constraint component to preserve latent relational information. Our model could perform patent classification and yield patent representation simultaneously. Compared with state-of-the-art baselines, patent2vec achieved significant performance improvement, which proved the potential of multi-view graph analysis on patent classification task. Extensive experimental discussions and case studies showed a significant improvement in accuracy and interpretability of our model. We hope our work will lead to more future studies in the patent mining field.

Acknowledgments This research was partially supported by grants from the National Key Research and Development Program of China (Grant No.2018YFB1402600), and the National Natural Science Foundation of China (Grant No.62072423).

References

1. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: CIKM 2015, pp. 891–900 (2015)
2. Chandra, D.K., Wang, P., Leopold, J., Fu, Y.: Collective representation learning on spatiotemporal heterogeneous information networks. In: SIGSPATIAL, pp. 319–328 (2019)
3. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
4. Dai, L., Yin, Y., Qin, C., Xu*, T., He, X., Chen, E., Xiong, H.: Enterprise Cooperation and Competition Analysis with Sign-Oriented Preference Network. In: Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD’20), pp. 774–782, San Diego, CA, USA (2020)
5. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: SIGKDD, pp. 135–144 (2017)
6. Evgeniya, U., Yaroslav, G., Victor, L.: Multi-region bilinear convolutional neural networks for person re-identification. In: AVSS, pp. 1–6. IEEE (2017)
7. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR, pp. 1933–1941 (2016)
8. Grawe, M.F., Martins, C.A., Bonfante, A.G.: Automated patent classification using word embedding. In: ICMLA, pp. 408–411. IEEE (2017)
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD, pp. 855–864 (2016)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS, pp. 1024–1034 (2017)
11. He, C., Loh, H.T.: Pattern-oriented associative rule-based patent classification. Expert Syst. Appl. **37**(3), 2395–2404 (2010)
12. Hu, J., Li, S., Hu, J., Yang, G.: A hierarchical feature extraction model for multi-label mechanical patent classification. Sustainability **10**(1), 219 (2018)
13. Hu, J., Li, S., Yao, Y., Yu, L., Yang, G., Hu, J.: Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy **20**(2), 104 (2018)

14. Jain, H., Prabhu, Y., Varma, M.: Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In: SIGKDD, pp. 935–944 (2016)
15. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2016)
16. Karpathy, A.ndrej., Li, F.ei.-F.ei.: Deep visual-semantic alignments for generating image descriptions. In: CVPR, pp. 3128–3137 (2015)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
18. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
19. Lai, K.-K., Wu, S.-J.: Using the patent co-citation approach to establish a new patent classification system. Inf. Process. Manage. **41**(2), 313–330 (2005)
20. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: ICML, pp. 1188–1196 (2014)
21. Lee, J.-S., Hsiang, J.: Patent classification by fine-tuning bert language model. World Patent Inf. **61**, 101965 (2020)
22. Li, S., Hu, J., Cui, Y., Hu, J.: Deeppatent: patent classification with convolutional neural networks and word embedding. Scientometrics **117**(2), 721–744 (2018)
23. Li, P., Xie, J., Wang, Q., Gao, Z.: Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In: CVPR, pp. 947–955 (2018)
24. Li, Y., Yang, M., Zhang, Z.: A survey of multi-view representation learning. IEEE TKDE **31**(10), 1863–1883 (2018)
25. Louay, A., Peter, K., Erdan, G., Stefan, F., Frank, H.: Optimizing neural networks for patent classification. In: ECML PKDD, pp. 688–703. Springer (2019)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
27. Nagrani, A., Albanie, S., Zisserman, A.: Learnable pins: Cross-modal embeddings for person identity. In: ECCV, pp. 71–88 (2018)
28. Peng, Y., Qi, J.: Cm-gans: Cross-modal generative adversarial networks for common representation learning. TOMM **15**(1), 1–24 (2019)
29. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD (2014)
30. Prabhu, Y., Varma, M.: Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In: SIGKDD, pp. 263–272 (2014)
31. Risch, J., Krestel, R.: Domain-specific word embeddings for patent classification. Data Technologies and Applications (2019)
32. Roudsari, A.H., Afshar, J., Lee, C.C., Lee, W.: Multi-label patent classification using attention-aware deep learning model. In: IEEE BigComp, pp. 558–559. IEEE (2020)
33. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
34. Smith, H.: Automation of patent classification. World Patent Inf. **24**(4), 269–271 (2002)
35. Tang, P., Jiang, M., Xia, B.(.Ning)., Pitera, J.W., Welser, J., Chawla, N.V.: Multi-label patent categorization with non-local attention-based graph convolutional network. In: AAAI, pp. 9024–9031 (2020)
36. Tang, J., Meng, Q., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
37. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
38. Wang, W., Arora, R., Livescu, K., Srebro, N.: Stochastic optimization for deep cca via nonlinear orthogonal iterations (2016)
39. Wang, P., Fu, Y., Xiong, H., Li, X.: Adversarial substructured representation learning for mobile user profiling. In: SIGKDD, pp. 130–138 (2019)
40. Wang, P., Fu, Y., Zhang, J., Wang, P., Yu, Z., Aggarwal, C.: You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis. In: SIGKDD, pp. 2457–2466 (2018)
41. Wang, P., Fu, Y., Zhou, Y., Liu, K., Li, X., Hua, K.: Exploiting mutual information for substructure-aware graph representation learning. In: IJCAI, pp. 3415–3421 (2020)
42. Wang, P., Li, X., Zheng, Y., Aggarwal, C., Fu, Y.: Spatiotemporal representation learning for driving behavior analysis. A joint perspective of peer and temporal dependencies. TKDE (2019)
43. Hao Wang, Tong Xu*, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, Wen Su: MCNE: An End-to-End Framework for Learning Multiple Conditional Network Representations of Social Network. In: Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'19), pp. 1064-1072, Anchorage, AK, USA (2019)

44. Wu, C.-H., Ken, Y., Huang, T.: Patent classification system using a new hybrid genetic algorithm support vector machine. Appl. Soft Comput. **10**(4), 1164–1177 (2010)
45. Xia, B., Baoan, L.I., Lv, X.: Research on patent document classification based on deep learning. In: AIIE. Atlantis Press (2016)
46. Zhang, L., Li, L., Li, T.: Patent mining: a survey. ACM SIGKDD Explorations Newsletter **16**(2), 1–19 (2015)
47. Zhang, L., Xu, T., Zhu, H., Qin, C., Meng, Q., Xiong, H., Chen, E.: Large-Scale Talent Flow Embedding for Company Competitive Analysis. In: Proceedings of The Web Conference 2020 (WWW'20), pp. 2354–2364, Taipei, China (2020)
48. Zhang, D., Liu, J., Zhu, H., Liu, Y., Wang, L., Wang, P., Xiong, H.: Job2vec: Job title benchmarking with collective multi-view representation learning. In: CIKM, pp. 2763–2771 (2019)
49. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR **9**(Nov), 2579–2605 (2008)

Affiliations

Lintao Fang[1] · Le Zhang[1] · Han Wu[1] · Tong Xu[1] · Ding Zhou[1] · Enhong Chen[1]

Lintao Fang
flt@mail.ustc.edu.cn

Le Zhang
zhangle0202@gmail.com

Han Wu
wuhanhan@mail.ustc.edu.cn

Ding Zhou
zhoudinglive@foxmail.com

Enhong Chen
cheneh@ustc.edu.cn

[1] Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China, Hefei, China