Linked Document Classification by Network Representation Learning

Yue Zhang[1,2], Liying Zhang[1,2] and Yao Liu[1*]

[1] Institute of Scientific and Technical Information of China, Beijing, China
[2] School of Software and Microelectronics, Peking University, Beijing, China
{zhangyuejoslin, zhangliying}@pku.edu.cn
liuy@istic.ac.cn

Abstract. Network Representation Learning (NRL) can learn a latent space representation of each vertex in a topology network structure to reflect linked information. Recently, NRL algorithms have been applied to obtain document embedding in linked document network, such as citation websites. However, most existing document representation methods with NRL are unsupervised and they cannot combine NRL with a concrete task-specific NLP tasks. So in this paper, we propose a unified end-to-end hybrid Linked Document Classification (LDC) model which can capture semantic features and topological structure of documents to improve the performance of document classification. In addition, we investigate to use a more flexible strategy to capture structure similarity to improve the traditional rigid extraction of linked document topology structure. The experimental results suggest that our proposed model outperforms other document classification methods especially in the case of having less training sets.

Keywords: Document Classification, NRL, Flexible Random Walk Strategy

1 Introduction

Document classification is a very prevalent topic in the field of NLP, and there have been quite a lot of research results, such as the combination of SVM classifier and rule-based classifier (Prabowo and Thelwall, 2009), the combination of Dependency Trees and CRF model (Nakagawa et al., 2010), and the ordinary BP neural network classification methods (Trappey et al., 2006). In general, the core of document classification is how to extract the key features of the text, and capture the mapping of features to categories.

The semantic features obtained by word2vec (Mikolov et al., 2013) and doc2vec (Le and Mikolov, 2014) are commonly used in document classification, and those algorithms usually assume that documents are independent of each other. But in linked document networks, such as citation websites, documents are inherently connected (citation relationship), and this network structure information has been proven useful for machine learning and data mining tasks (Massa and Avesani, 2007; Mei et

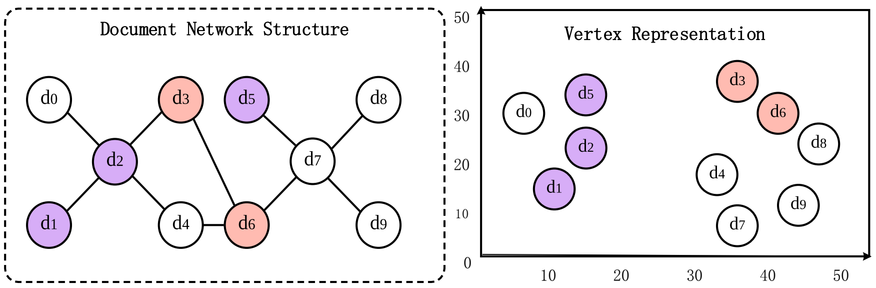

Fig. 1. Document Network Representation Learning Illustration by Deepwalk

al., 2009; Tang and Liu, 2012). Therefore, in addition to the semantic features, the topology structure features among documents should also be considered in linked document classification. To join the linked features to documents, some researchers have investigated to introduce the network representation learning algorithm into language models, such as LDE (Wang et al., 2016) and Tri-party (Pan et al., 2016). Figure 1 illustrates the document network representation learning by deepwalk (Perozzi et al., 2014). Deepwalk is one of the traditional algorithms utilized to capture the local structure information of the vertices in a network. In Figure1, each vertex denotes a paper, and the links are citation relationships. Different colors are the classification labels of documents, and the rest are unlabeled. Among them, d0 and d2 are connected, and then their vertex two-dimensional vector representations are close together even the label of d0 is unknown. However, deepwalk is not expressive enough to capture the diversity of connectivity patterns observed in networks because it can only obtain interconnected document information but lose identical structure information. For instance, as for d0 and d7, although they are not strictly connected, they share the same structural role in a hub document. So in theory, the distributed representation of d0 and d7 should also be close to each other, but according to the illustration, it's evident that deepwalk algorithm can't capture this kind of information.

Hence, in this paper, we study the novel problems of document classification based on network representation mainly from two specific aspects: (1) How to jointly learn document embedding and document topological structure, and apply the mixed feature into classification. (2) How to get more comprehensive and flexible document network embedding. Based on those problems, we propose a linked document framework for classification (LDC). The primary contribution of this paper is as follows:

(1) We provide a unified end-to-end model (LDC) to learn document semantic content and document topological structure jointly to improve the effectiveness of document classification.

(2) We get document network representations not only from the documents which are interconnected but also the documents which share similar topological structures. It improves the performance of document classification, especially in the case of small training sets.

(3) We evaluate our approach using DBLP[1]and Cite-Seer-M10[2] datasets. The results demonstrate the advantages of our method compared with 5 baselines.

2 Related Work

The proposed model in this paper is based on network representation learning (network embedding). It aims to encode each vertex in the network as low-dimensional and dense vector, which can be easily and conveniently used as input in a machine learning model. Furthermore, the obtained vertex representation can be applied to common applications, such as visualization task, node classification, link prediction and community discovery.

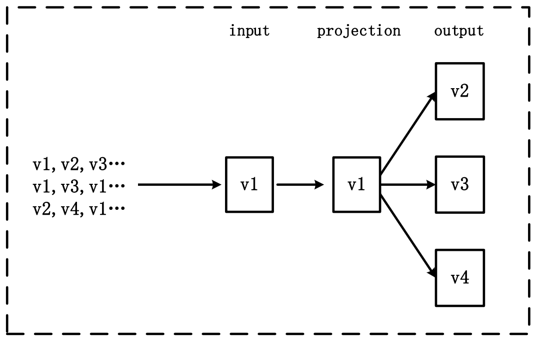

Fig. 2. Architecture of Deepwalk Algorithm

Deepwalk algorithm mentioned above is one of the most commonly used NRL algorithms. It introduces the skip-gram algorithm into NRL and employs word2vec model to embed all nodes into a continuous vector space. The architecture of deepwalk algorithm has been shown in Figure2. Each vertex in random walk such as $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n$ can be considered as a word, and their random walk can be viewed as sentences. Then those sentences are input into word2vec model, which in turn yields the representation of each vertex.

Recently, some researchers have begun to introduce NRL models to NLP tasks. Yang proposed the text-associated deepwalk (TADW) model, taking into account the structure information and content information of the vertex, and combining the vertex text features into network representation learning under the framework of the matrix decomposition (Yang et al., 2015). Similar to TADW, such representation models based on network structure-content fusion learning include MFR (Li et al., 2015), Author2Vec (Genash J, et al., 2016), and etc. In addition to structure-content modeling, some algorithms even take document label information into consideration. Wang proposed the LDE model, employing distributional hypothesis idea for document

[1] http://arnetminer.org/citation (V4 version is used)
[2] http://citeseerx.ist.psu.edu/

embedding by combining link and label information with content simultaneously (Wang et al, 2016). Pan proposed the TriDNR model, Tri-party Deep Network Representation, exploited inter-node relationships, node-content correlation, and label-content correspondence in a network to learn representation for each vertex (Pan et al., 2016).

The similarity of those model is that they are unsupervised. Although LDE and TriDNR consider label information, they treat label as one of the features. So we put forward a combined model to incorporate task-specific supervision, which means to use label information supervise text and structure learning process simultaneously. The model improves the performance of text classification, at the same time, the intermediate training parameters of each vertex can better task-specific representation.

3 Model

3.1 Problem Statement

We first formally define the notation in linked document classification by network representation learning. Let $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{D})$ denote a document network with texts and labels in each vertex, where $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2 \cdots \mathrm{v}_\mathrm{N}\}$ is a set of connected N documents and v_i is the i-th vertex in the network. $\mathrm{e}_{\mathrm{i,j}} = (\mathrm{v}_\mathrm{i}, \mathrm{v}_\mathrm{j}) \in \mathrm{E}$ represents the edge relationship between v_i and v_j, $\mathrm{D} = \{\mathrm{d}_1, \mathrm{d}_2, \cdots \mathrm{d}_\mathrm{N}\}$ is the text information and $\mathrm{d}_\mathrm{i} = \{\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_\mathrm{n}\}$ is the word representation of the i-th document. The primary task in this paper is to classify connected documents by learning the feature of the text and the topology structure jointly.

3.2 LDC

We propose a unified end-to-end hybrid model (LDC) to jointly capture the semantic and topology structure of linked document network. The overview of our model has been illustrated in Figure3. The overall model is divided into three major parts, which are semantic feature modeling, topological structure modeling and fusion modeling.

The inputs of the model are text content and linked information of documents. Among them, the word embedding is obtained from Glove which is a count-based model utilizing word co-occurrence matrix (Pennington et al., 2014), and the node embedding is trained by the node2vec algorithm (Grover et al., 2016).

We develop two deep neural network architectures, which are Convolution Neural Network (CNN) and Deep Neural Network (DNN). The input of word embedding is fed into CNN model to extract meaningful local semantic features in the text. The input of node embedding is fed into DNN model (two-layer) to get more abstract and smooth features in the document network structure.

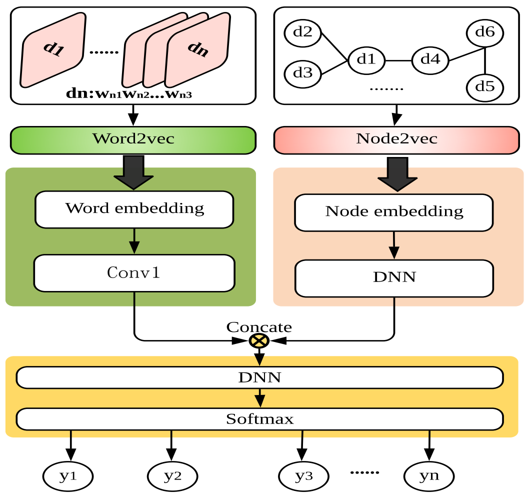

Fig. 3. LDC architecture. Conv: Convolution; DNN: Deep Neural Network.

After concating word and structure high-level features, the mixed representation is put into another fully four-layer connected deep neural network. The final softmax layer is to utilize classification label information to supervise the whole learning process. Through jointly learning the features of topology structure and document text, a better performance of text classification can be achieved.

3.3 Semantic Feature Modeling

There has been a variety of models to obtain document embedding from the word sequence. The traditional unsupervised algorithms, such as doc2vec (Micolov et al., 2013), are not able to consider classification label information during the training process. Some tasks (Pan et al., 2016) consider label after a classifier is trained with document representations. However, experiment results prove that unsupervised text representation methods usually yield inferior results especially in particular machine learning tasks (Tang J et al., 2015). In this section, we investigate different neural networks with label information for text modeling including CNN (Kalchbrenner et al., 2014), RNN (Cho et al., 2014), and Bidirectional RNN (Schuster and Paliwal, 1997). The result is that CNN performs best as it can capture the local semantic dependency among words (Tu et al., 2017) by convolution and max-pooling layers. The input of CNN model is the word sequence representation of each document, and then CNN model gets document embedding though three layers, i.e. embedding layer, convolution layer and max-pooling layer.

Embedding Layer. The underlying assumption of word2vec is that “you shall know a word by the company it keeps,” (Li J et al., 2016). In a word, a sound word representation should be used in predicting its nearby words. Word representation layer transfers the word sequence $D = (w_1, w_2, \dots, w_n)$ in each document to corresponding word embedding. For instance, a sentence of length n (has been zero padded) can be represented as:

$$w_{1:n} = w_1 \oplus w_2 \cdots \oplus w_n \qquad (1)$$

Where $\oplus$ is the concatenation operator. $w_{i:j}$ is the concatenation of words $w_i, w_{i+1}, \cdots w_{i+j}$.

Convolution Layer. After embedding layer, convolution layer can extract local features of word embedding of S. The convolution kernel is a sliding window of h words with a convolution matrix $K \in \mathbb{R}^{d\times(h\times d'')}$. Then:

$$Z_i = K \times S_{i:i+h-1} + b \qquad (2)$$

Where $S_{i:i+h-1}$ is the concatenation of word embedding within the i-th window and b is the bias vector.

Global Max-Pooling Layer. The vector obtained after convolution layer is combined into a vector sequence $(x_0^i, \cdots, x_n^i)$. In order to capture the most important information related to tasks, we employ the global max-pooling and non-linear transformation as following:

$$d_i = \tanh(\max(x_0^i, \cdots, x_n^i)) \qquad (3)$$

At last, we encode the text information of each document as $d^t = [d_1, \cdots, d_n]^T$. The document embedding trained by CNN model not only acquires the rich local features, but also embeds the label information into its own representation.

3.4 Topology Structure Feature Modeling

If the semantic feature of the text is not apparent, for instance, there are no same surrounding words between texts, then the topological structure feature should be used to indicate document's closeness in document network. As mentioned above, deepwalk is proposed as a method to learn network graphs based on skip-gram, and it employs the language model word2vec to learn a latent representation of each vertex. Mathematically, we simulate a random walk of fixed length *l* with a source node $t = c_0$ then the distribution of i-th node v_i in the walk is following:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{z} & \text{if}(v, x)\epsilon E \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Where π_{vx} is the normalized transition probability between nodes v and x, and z is the normalizing constant. The assumption of deepwalk is that interconnected documents are more likely to be in the same category. However, this assumption is hidebound because documents could be far apart in the network but still have the same structural role, which also can indicate the same category. So we adjust the random walk strategy by utilizing document topology structure feature itself.

Rather than the rigid defining neighborhood for each vertex, the node2vec algorithm designs a flexible neighborhood strategy. It obtains a more comprehensive vertex representation mainly from two aspects, which are homophily and structural equivalence. In terms of homophily, vertices that are highly interconnected should be embedded close together. In contrast, in terms of structural equivalence, the documents sharing the same structural role should also be embedded close together even without interconnection. Among them, the homophily and structural equivalence are controlled by Breadth-first Sampling (BFS) and Depth-first Sampling (DFS) respectively. The concrete method is to multiple biases α to the weight of edges, which can be calculated through linked probability. Then π_{vx} in node2vec is:

$$\pi_{vx} = \alpha_{pq}(v, x) \cdot w_{vx} \tag{5}$$

Where w_{vx} is the weight of $E(v, x)$. Two parameters p and q are designed to adjust the random walk as follows:

$$\alpha_{pq}(v, x) = \begin{cases} \frac{1}{p} & \text{if } d_{vx} = 0 \\ 1 & \text{if } d_{vx} = 1 \\ \frac{1}{q} & \text{if } d_{vx} = 2 \end{cases} \tag{6}$$

Where d_{vx} is the shortest path between v and x. Assume the source vertex is t, and t moves to vertex x. If p>1 (DFS), the next walk of x will traverse biasedly the vertices away from t, while if q>1(DFS), the next random of x will traverse biasedly the vertices neighboring of t.

After getting the vertex representation, we did a comparative experiment. On one hand, we concate the node representation and document semantic representation directly; on the other hand, we feed the vertex representation into a deep neural network (two-layers) before concating with semantic representation. The experiment result shows that DNN performs better. The reason is that DNN can extract different levels of features and low-level features can be converted into a more abstract high-level representation through combination. In general, in section of topology structure feature modeling of documents, we not only use node2vec algorithm to get vertex representation, but also feed the representation into a DNN model to obtain a more abstract feature. The feature we get can be represented as $v^t = [v_1, \cdots, v_n]^T$.

3.5 Fusion

Given the semantic representation d^t and topology structure representation v^t, we can obtain the mix embedding as $u^t = d^t \otimes v^t$ .Then u^t is fed into fully connected layers (four-layers). Finally, the final softmax layer, which contains probability distribution over the label, supervises the training process of document semantic embedding and structure embedding simultaneously. Let $Y \in R^{d \times L_c}$ be the label embedding matrix, where L_c is the number of unique labels. Then:

$$\max_{Y, U} \frac{1}{|y|} \sum_{i: y_i \in y} \log P(y_{u_i} | u_i) \tag{7}$$

Where U represents the documents with the combination of semantic and topology structure features. y_i is the label of the i-th document. And $P(y_{u_i}|u_i)$ is the probability that u_i's label is y_{u_i}, which is given as

$$P(y_{u_i}|d_i) = \frac{\exp(y_{u_i}^T u_i)}{\sum_{k=1}^{L_c} \exp(y_k^T u_i)} \quad (8)$$

The objective of LDC model is to minimize the following log-likelihood:

$$\zeta = \delta \sum_{i=1}^{N} \sum_{r \in R} \sum_{-t \le j \le t} \log P(v_{i+j}|v_i) + \mu \sum_{i=1}^{N} \sum_{-t \le j \le t} \log P(w_j|d_i) \quad (9)$$

The first term aims to learn vertex network structure information, and t is the window size, r is the random walks generated by the node2vec algorithm. The second term is to obtain the semantic information, and w_j is the j-th word in window size. δ and μ are the weights that balance network structure and text information.

4 Experiment Result

In order to investigate the effectiveness of LDC model, we conducted an experiment of document classification on two datasets. Both of them are paper citation networks.

4.1 Datasets

Table 1. Paper Citation Datasets

Datasets	DBLP	Cite-Seer-M10
#vertices	60744	10310
#edges	52890	77218
#group	4	10

DBLP is a computer science bibliography website. Each paper may cite or be cited by other papers and form a citation website. We selected 60744 papers and labeled them into 4 categories, which are about the database, AI, computer vision and data mining.

Cite-Seer-M10 is a subset of Cite-Seer-M10 data which consists of scientific publications. We selected 10310 papers and labeled them into 10 categories, which are about archaeology, financial economics, agriculture, biology, archeology, material science, industrial engineering, petroleum chemistry, physics and social science.

4.2 Evaluation Metrics and Experiment Settings

We use the standard metrics of classification, Average Macro-F1, and Average Micro-F1. The larger the Macro-F1 and Micro-F1 are, the better the document representation is for the classification task. The default parameters for algorithms as follows: window size R=5, dimension b=300, training size p=70%, epoch=20, batch_size=100, $\delta = 1, \mu = 1$. For fairness, we set the same dimension and training size for all algorithms. The hidden units of DNN in semantic feature modeling are 300, and the hidden units of DNN in Fusion are 300,128, 64, and 32.

4.3 Experiment Performance Comparison

Table 2. Document classification comparison on datasets

Datasets	**DBLP**		**Cite-Seer-M10**	
Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1
SVM	0.635±0.002	0.612±0.003	0.655±0.006	0.632±0.005
LSTM	0.683±0.006	0.681±0.004	0.715±0.004	0.695±0.004
CNN	0.733±0.004	0.721±0.003	0.749±0.008	0.752±0.003
d2v+dw	0.757±0.003	0.745±0.002	0.771±0.003	0.773±0.002
TriDNR	0.782±0.003	0.770±0.004	0.785±0.006	0.789±0.004
LDC	**0.810±0.005**	**0.813±0.006**	**0.821±0.002**	**0.819±0.005**

Table 2 shows the performance of different algorithms on two test datasets. The input of SVM, LSTM and CNN are the document embeddings obtained from d2v algorithm. The results show that the neural network methods perform better than traditional machine learning algorithms. d2v+dw represents the concatenation of the vector representation learned by doc2vec and deepwalk, and its representation is fed into a CNN model. d2v+dw has better results because it combines semantic and structural information at the same time. Furthermore, we utilize TriDNR algorithm to get each vertex representation and input it into an SVM classifier, the results show that it performs better than d2v+dw because the label information is considered. But TriDNR is still far from optimal, comparing with LDC algorithm. To verify the effect of different dimensions of representation on the classification effect, we verify dimension b increasing from 100 to 300, Figure4 shows the variation of Macro-F1 with different dimensions and there is a slight rise for LDC, which proves LDC is quite stable.

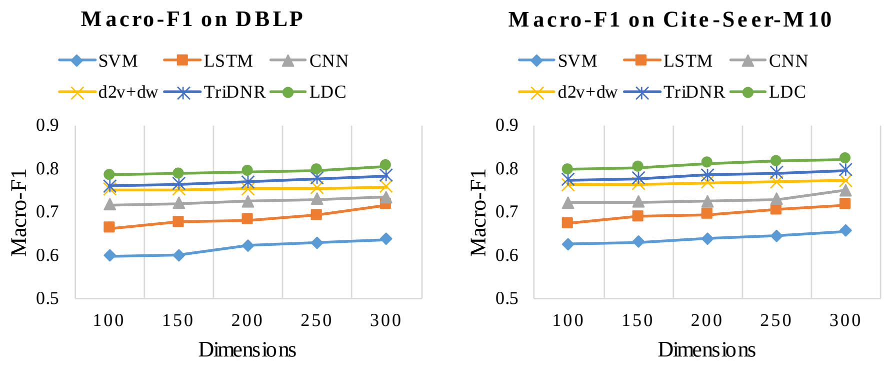

Fig. 4. Experiments with different dimensions

At the same time, we vary the percentage of training sets p from 10% to 70% on DBLP datasets, and we change different network representation learning algorithms. Table 3 is the results. As for node2vec, the best exploration strategy (p=0.25, q=0.5) turns out to perform better than deepwalk (p=1, q=1) and LINE. Especially under the circumstance of 10% training sets. The reason is that when we have small training sets, the influence of feature of semantic would be little, which leads to the feature of topology structure is more apparent. When we lower the parameter of p and q, the feature of topological structure can be adequately captured from both macro and micro views.

Table 3. Average macro-F1 score on DBLP datasets

%p	deepwalk	Line	node2vec
10	64.3%	58.4%	71.1%
30	72.2%	66.2%	73.5%
50	77.3%	68.2%	79.2%
70	79.4%	70.2%	80.9%

4.4 Case

As we mentioned before, the intermediate training parameters of each vertex in LDC can be treated as task-specific representation. For the purpose of validating this conclusion, we visualize the DBLP document representation (4 categories, 100 dimensions) in 2D space in Figure5 by Google Embedding Projector (Smikov and Daniel et al.,2016), which is a tool to visualize high-dimension data. The dimension reduction method we choose is custom linear projection. It can help discover useful direction in the dataset, such as the difference between a formal tone and an informal tone in a language generation model.

According to the results, the category boundaries of deepwalk are very blurred, which indicates that network feature alone is not apparent. TriDNR performs much better than deepwalk because it combines semantic, structural and label information. However, TriDNR is still slightly overlapping among four categories. The result of LDC is quite clear, which proves that our proposed supervised model can yield better task-specific linked document representation than other methods.

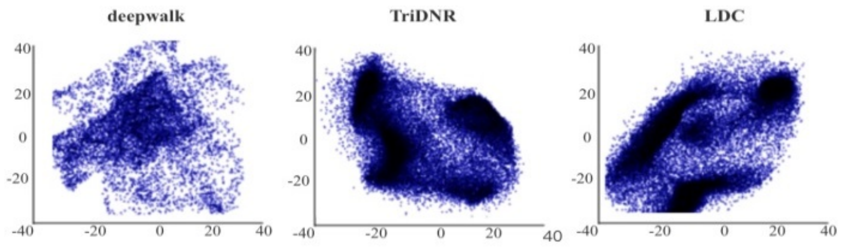

Fig. 5. Two-dimension visualization of DBLP

5 Conclusion

This paper presents a unified end-to-end LDC model, a new approach for document classification on citation networks. We not only propose a method learning text information and network topology structure jointly, but also improve the flexibility of random walk strategies by learning the document network structure feature itself. LDC gains significantly better performance than other document classification baseline methods. Same as other document classification methods, LDC can be used not only in classification, but also in sentiment analysis and information retrieval.

The future work has two main directions: (1) We will expand our investigation to heterogeneous network application in the language model. (2) We have proved the effectiveness of LDC on document embedding for classification, and we will investigate its performance on other specific tasks such as ranking or recommendation.

References

1. Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent dirichlet allocation. the Journal of machine Learning research, 3:993-1022 (2003).
2. Cai, D., Zhang, D., Zhai, C.: Topic modeling with network regularization. In: Proceedings of the 17th international conference on World Wide Web, pp. 101-110. ACM, China (2008).
3. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014).
4. Ganguly, S., Gupta, M., Varma, V., Pudi, V.: Author2Vec: Learning Author Representations by Combining Content and Link Information. International Conference Companion

on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 49-50 (2016).

5. Grover, A., Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855-864. ACM (2016).
6. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014).
7. Landauer, T. K., Foltz, P. W., Laham, D.: An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284 (1998).
8. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188-1196 (2014).
9. Li, J., Ritter, A., Jurafsky, D.: Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. arXiv preprint arXiv:1510.05198 (2015).
10. Li, J., Zhu, J., Zhang, B.: Discriminative Deep Random Walk for Network Classification. Meeting of the Association for Computational Linguistics, pp. 1004-1013 (2016).
11. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems, pp. 17-24. ACM (2007).
12. Mei, Q. Ma, H., Lyu, M. R., King, I.: Learning to recommend with trust and distrust relationships. In RecSys, pp. 189-196. ACM (2009).
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
14. Nakagawa, T., Inui, K., Kurohashi, S.: Dependency tree-based sentiment classification using CRFs with hidden variables. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 786-794. Association for Computational Linguistics (2010).
15. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. International Joint Conference on Artificial Intelligence, pp. 1895-1901. AAAI Press (2016).
16. Perozzi, B., Al-Rfou, R., & Skiena, S.: Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, (pp. 701-710. ACM (2014).
17. Pennington, J., Socher, R., Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing. EMNLP, pp. 1532-1543 (2014).
18. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. Journal of Informetrics, 3(2), 143-157 (2009).
19. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5):513-523 (1988).
20. Schuster, M., Paliwal, K. K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673-2681 (1997).
21. Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F. B., Wattenberg, M. Embedding Projector: Interactive visualization and interpretation of embeddings. arXiv preprint arXiv:1611.05469 (2016).
22. Sun, X., Guo, J., Ding, X., Liu, T.: A General Framework for Content-enhanced Network Representation Learning. arXiv preprint arXiv:1610.02906 (2016).
23. Tang, J., Liu, H.: Feature selection with linked data in social media. In: Proceedings of the 2012 SIAM International Conference on Data Mining, pp. 118-128. Society for Industrial and Applied Mathematics (2012).

24. Tang, J., Qu, M., Mei, Q.: 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 (2015).
25. Trappey, A. J., Hsu, F. C., Trappey, C. V., Lin, C. I.: Development of a patent document classification and search platform using a back-propagation network. Expert Systems with Applications, 31(4), 755-765 (2006).
26. Tu, C., Liu, H., Liu, Z., Sun, M.: Cane: Context-aware network embedding for relation modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vol. 1, pp. 1722-1731 (2017).
27. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E. Y.: Network representation learning with rich text information. International Conference on Artificial Intelligence. AAAI Press, pp. 2111-2117 (2015).
28. Wang, S., Tang, J., Aggarwal, C., Liu, H.: Linked document embedding for classification. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 115-124. ACM (2016).
29. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network Representation Learning: A Survey. arXiv preprint arXiv:1801.05852 (2017).