Coping for missing content and missing links in document network embedding

Jean 2020 Dupuy

Student Research Abstract: Coping for Missing Content and Missing Links in Document Network Embedding

Jean Dupuy
Université de Lyon, Lyon 2, ERIC EA3083 – MeetSYS
Lyon, France
jean.dupuy@univ-lyon2.fr

ABSTRACT

Searching through networks of documents is an important task. A promising path to improve the performance of information retrieval systems in this context is to leverage dense node and content representations learned with embedding techniques. However, these techniques cannot learn representations for documents that are either isolated or whose content is missing. To tackle this issue, assuming that the topology of the network and the content of the documents correlate, we propose to estimate the missing node representations from the available content representations, and conversely. Inspired by recent advances in machine translation, we detail in this paper how to learn a linear transformation from a set of aligned content and node representations. The projection matrix is efficiently calculated in terms of the singular value decomposition. The usefulness of the proposed method is highlighted by the improved ability to predict the neighborhood of nodes whose links are unobserved based on the projected content representations, and to retrieve similar documents when content is missing, based on the projected node representations.

CCS CONCEPTS

• **Computing methodologies** → **Machine translation**; *Learning latent representations*; • **Information systems** → *Information retrieval*;

KEYWORDS

Document network, Document representation, Node representation, Translation

ACM Reference Format:
Jean Dupuy. 2020. Student Research Abstract: Coping for Missing Content and Missing Links in Document Network Embedding. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC'20), March 30-April 3, 2020, Brno, CzechRepublic.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341105.3374222

1 INTRODUCTION

Searching through a network of documents, such as scientific articles [2], Web pages or social media posts [5], is a common task and a long-standing research topic. Recent works propose to improve the performance of information retrieval systems by leveraging dense vector space representations of the content and the nodes [16, 19]. Quite often, for various reasons, such as technical issues or legal limitations (*e.g.* paywall), not all links can be observed nor can all the content be retrieved.

These representations are learned with algorithms similar to those devised for word embedding, *e.g.* DeepWalk [14], Node2Vec [4] or GVNR [3] for node representations, or Doc2Vec [7] for content. Obviously, missing content prevents applying any document embedding techniques. Also, since network embedding algorithms perform random walks to measure the co-occurrence frequency between nodes, they cannot learn representations of isolated nodes.

In this paper, we address this issue by formulating a translation task, from available content representations to node representations, and from available node representations to content representations. Assuming that the topology of a network of documents correlates with the content of the documents, we propose to estimate the missing node (resp. document) representations as a linear transformation, *i.e.* projection, of the content (resp. node) representations. More specifically, we show how to construct a dictionary of pairs of content and node representations that allows learning a projection matrix, inspired by recent advances in machine translation with pre-trained word embeddings [6, 12, 18]. Note that a self-translation network embedding algorithm, STNE [9], has been recently proposed, however it addresses a different task. It learns to translate a sequence of content to a sequence of nodes to learn richer node representations and it cannot operate on isolated nodes nor deal with missing content.

Our approach proves relevant, as evidenced by the quality of the alignment between the estimated representations and the true representations. Practically speaking, our method has two main applications. On the one hand, estimating the node representation helps in recovering the missing links. On the other hand, estimating the content representation from the node representation allows measuring the similarity with the content of other documents and thus provides helpful clues about the missing content.

The rest of this paper is organized as follows. In section 2, we survey related work on document and network embedding, as well as embedding-based bilingual translation. We then describe our proposal to estimate the missing representations in section 3. Next, in section 4, we evaluate the quality of the projection and assess the improvement brought by our method in predicting the neighborhood of nodes whose links are missing and retrieve documents similar to those without content. Finally, we conclude and provide future directions in section 5.

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03.
https://doi.org/10.1145/3341105.3374222

2 RELATED WORK

Word embedding, *i.e.*, the task of learning dense vector space representations of words, is tightly connected to the tasks of learning document and node representations, *i.e.* document embedding and network embedding. In this section, we first briefly survey works related to word embedding and then present models adapted for document and network embedding. Lastly, we cover recent advances in offline bilingual translation via word embeddings, on which our work is based.

2.1 Word Embedding

Word embedding builds upon the distributional hypothesis. It states that distributional similarity and meaning similarity are correlated, which allows learning representations of words based on the contexts in which they occur, the context of a word being co-occurring words in a short window.

Mikolov *et al.* [11] propose two models, namely Skip-Gram and Continuous Bag-of-words (CBOW). The first models the conditional probability of occurrence of a word, given a word in the context. The second models the conditional probability of occurrence of a word, given the set of words in the context. In both cases, the probability is defined as a softmax function parameterized by the dot products of the vector representations of the words.

Because the softmax makes the estimation of the word representations computationally expensive, Mikolov *et al.* [13] propose approximate solutions based either on the hierarchical softmax or negative sampling, an approximation akin to noise-contrastive estimation.

2.2 Content Representation Learning

Le and Mikolov [7] extends the Skip-Gram and CBOW models to learn representations of documents, under the names, respectively, Distributed Bag-of-words (DV-DBOW) and Distributed Memory (DV-DM). They rely on a simple trick, that consists in considering each document as a distinct new vocabulary entry, that occurs in every window in the document. Thus, these models learn document representations that are good at predicting the words they contain.

2.3 Node Representation Learning

Even though the distributional hypothesis originated in linguistics, Perozzi *et al.* [14] show that sequences of nodes generated by random walks and sentences share some statistical properties. In particular, they show that the frequency at which nodes appear in short random walks follows a power-law distribution, like the frequency of words in language. Consequently, they propose to learn node representations under the distributional hypothesis. The proposed model, DeepWalk, consists in learning the representations based on the Skip-Gram model with the hierarchical softmax approximation, from a corpus of node sequences – deemed equivalent to sentences – generated by truncated random walks.

LINE [17] refines the objective optimized by DeepWalk, trying to preserve both the first and second order proximities in the embedding space. The representations are learned based on the Skip-Gram model with negative sampling rather than the hierarchical softmax approximation, using a specific stochastic gradient descent algorithm for weighted networks, that samples the edges according to their weights.

Node2vec [4] performs biased random walks, in order to better balance the exploration-exploitation trade-off, arguing that the added flexibility in exploring neighborhoods helps learning richer representations. The representations are also learned based on the Skip-Gram model with negative sampling.

2.4 Bilingual Translation

Mikolov *et al.* [12] first propose to learn a linear transformation from the embedding space of a source language into the embedding space of a target language. They suggest to estimate a projection matrix from pre-trained vectors and a bilingual dictionary. To this end they formulate a least square problem based on the Euclidean distance between aligned vectors, which they solve with the regular stochastic gradient descent algorithm.

Even though this approach yields encouraging results, Xing *et al.* [18] argue that the problem is ill-posed, since the similarity between representations is usually measured in terms of the cosine similarity, rather than the Euclidean distance. They show how to learn unitary word embeddings and formulate a maximization problem in terms of the dot product between some source vectors and projected target vectors. They also require the projection matrix to be orthogonal, so that the projected vectors remain unitary in order to preserve the equivalence between the dot product and the cosine similarity. Eventually, they devise a gradient descent based algorithm to learn the projection matrix. Yet, their approach is computationally expensive, because it requires to orthogonalize the projection matrix after each update, by computing its singular value decomposition.

Smith *et al.* [6] show that finding the projection matrix that solves the problem stated by Xing *et al.* is actually equivalent to solving a least square problem based on the Euclidean distance, when the vectors are unitary and the projection matrix is orthogonal. Furthermore, they link this problem with the orthogonal Procrustes problem, which admits an analytical solution. The projection matrix is obtained from the singular value decomposition of the similarity matrix between the aligned source and target vectors.

3 PROPOSED METHOD

In this section, we describe our proposal, which consists in adapting techniques devised for bilingual translation to learn a projection from the content representations (pre-trained with, *e.g.* Doc2Vec) to the node representations (pre-trained with, *e.g.* DeepWalk), and vice versa.

Consider a corpus of documents organized in a network, so that the content of some documents is missing and the links of some other documents are unobserved. We assume a set of pre-trained node representations $A = \{a_i \in \mathbb{R}^k\}$ and a set of document representations $B = \{b_i \in \mathbb{R}^k\}$. Our goal is to estimate the missing node representations in A, from the content representations in B ; conversely, estimate the missing content representations in B, from the node representations in A.

We posit that the topology of the network and the content of the documents correlate. Thus, we aim at finding a projection matrix

$W \in \mathbb{R}^{k \times k}$ that maps content representations to node representations, and node representations to content representations. More particularly, given a document i, we want W to maximize the cosine of the angle $\theta_{a_i, b_i W}$ between a_i and $b_i W$. That is to say, we want to find a linear transformation so that the vectors a_i and $b_i W$ point in the same direction. More formally, based on a set of pairs of node representations and document representations, S, we want to find a matrix W so that:

$$\max_W \sum_{i \in S} \cos(\theta_{a_i, b_i W}), \tag{1}$$

In the rest of this section, we describe how to construct S and then detail how to learn the projection matrix W.

3.1 Dictionary construction

Rather than learning the projection from all the available pairs of node representations and content representations, we suggest to restrain to a specific subset. The aim in doing so is (i) favor the learning of a good projection and (ii) limit the computational cost.

Bojanowski *et al.* [1] note that Skip-Gram, the model behind most network embedding algorithms, struggles to learn good representations for rare words. This also holds for network embedding algorithms, since Perrozi *et al.* [14] show that the frequency at which nodes occur in short random walks and the frequency of words in language follow a similar law. Hence, we propose to avoid rare nodes and to construct the dictionary only from the m nodes that occur the most in short random walks on the network. Formally, with f_i the frequency of node i, we construct the set S of cardinality m that maximizes $\sum_{i \in S} f_i$.

3.2 Projection Learning

Following [18], we rewrite formula 1 in terms of the dot product, and thus require the node and content representation to be unitary vectors and constrain W to be orthogonal:

$$\max_W \sum_{i \in S} a_i \cdot b_i W, \text{ subject to } W^\top W = I. \tag{2}$$

Yet, rather than learning the representations again as Xing *et al.* suggest, we simply row-normalize A and B. Then, thanks to the proportional relationship between the opposite of the dot product and the squared Euclidean distance highlighted in [6], because $||a_i - b_i W||^2 = ||a_i||^2 + ||b_i||^2 - 2(a_i \cdot b_i W)$, we transform the problem into a minimization one:

$$\min_W \sum_{i \in S} ||a_i - b_i W||^2, \text{ subject to } W^\top W = I. \tag{3}$$

This is a classic linear algebra problem, which solution is given in [15]. It consists in calculating the singular value decomposition of the matrix $M = A_S^\top B_S$, with $A_S \in \mathbb{R}^{m \times k}$ and $B_S \in \mathbb{R}^{m \times k}$ the aligned node and content representations according to the dictionary S. With $M = U \Sigma V^\top$, the matrix W is defined as:

$$W = U V^\top. \tag{4}$$

We estimate the node representation of an isolated node by calculating $\tilde{a} = bW$. Similarly, we estimate the content representation of a document whose content is missing by calculating $\tilde{b} = aW^\top$.

Table 1: Network properties.

Name	# of documents	# of links
Cora	2,211	5,001
HepTh	1,038	1,974

4 EXPERIMENTS

In this section, we evaluate our proposal on two networks of documents. First, we focus on missing links and show that estimating the node representations from the content representations allows to better reconstruct the neighborhood of nodes whose links are unobserved. Second, we focus on missing content and show that estimating the content representations from node representations allows to retrieve a significant fraction of the similar documents.

4.1 Experimental setup

4.1.1 Data. We consider two well known networks of scientific articles, Cora [10] and HepTh [8], whose properties are summed up in Table 1. No content nor citation links are missing. These datasets available online[1].

4.1.2 Node and Content Representations. We experiment with DeepWalk [14] to learn the node representations, with the following configuration: 80 random walks length 80 from each node, and a co-occurrence window of size 10. Regarding content representation, we experiment with the Distributed Memory (DV-DM) variant of Doc2Vec [7] with a window of size 10. Here after, we report the results for representations of size 500.

4.1.3 Projection Learning. We learn the projection matrix via the exact singular value decomposition, based on a dictionary of size 1400 for Cora, and a dictionary of size 550 for HepTh.

4.1.4 Similarity Metric. We measure the similarity between vector representations in terms of cosine similarity [11].

4.2 Missing Links: Neighborhood Reconstruction

We begin by assessing the quality of the estimated node representations. The evaluation task consists in reconstructing the neighborhood of a document whose links are unobserved. We adopt a leave-one-out strategy and proceed in the following manner. First, we learn all the content representations from the entire corpus. Then, for each document i, we do the following:

(1) Hide all the incoming and outgoing links of document i;
(2) Learn the node representations on this sub-network;
(3) Learn the projection matrix on this sub-network;
(4) Measure the similarity between the estimated node representation $\tilde{a}_i = b_i W$ and all the other node representations $\{a_j\}$;
(5) Order the documents accordingly and measure the precision at n, $P@n$ w.r.t. the true links.

[1] Get the data: https://github.com/thunlp/CANE/tree/master/datasets/cora, https://github.com/thunlp/CANE/tree/master/datasets/HepTh

Table 2: Precision gain in link prediction.

		P@5	P@10	P@20	P@50
2*Cora	$\cos(\theta_{b_i,b_j})$	+4.4%	+23.4%	+20.1%	+20.0%
	Random_{200}	+681.4%	+475.9%	+166.1%	+51.9%
2*Hepth	$\cos(\theta_{b_i,b_j})$	+69.2%	+42.9%	+5.9%	+2.6%
	Random_{200}	+516.6%	+375.1%	+350.4%	+175.6%

We report the relative gain in average precision in Table 2, w.r.t. two baselines:

- $\cos(\theta_{b_i,b_j})$: Documents ordered according to the similarity between the content representations;
- Random_{200}: Documents ordered randomly, picked among the 200 documents with the highest degrees.

It reveals that the estimated node representations, obtained via the linear transformation of the content representations always lead to a better precision in predicting the neighborhood of a document.

4.3 Missing Content: Similar Document Retrieval

We now switch our attention to missing content. We assess the quality of the estimated content representations following a similar leave-one-out methodology. We learn all the the content representations and define the true sets of similar document $S_i = \{j|\cos(\theta_{b_i,b_j}) > 0.2\}$. The evaluation task consists in reconstructing the set of similar documents for a document whose content is unknown. We proceed as follows. First, we learn all the node representations from the whole network. Then, for each document i, we do the following:

(1) Hide document i from the corpus;
(2) Retrain the content representations on this sub-corpus;
(3) Learn the projection matrix on this sub-corpus;
(4) Measure the similarity between the estimated content representation $\tilde{b_i} = a_i W^\top$ and all the content representations $\{b_j\}$;
(5) Measure the accuracy by comparing the true set of similar documents S_i and the estimated set of similar documents $\tilde{S_i} = \{j|\cos(\theta_{\tilde{b}_i,b_j}) > 0.2\}$.

We report the average precision following this procedure in the first row of Table 3. The second row shows the average precision based on the node representations, where $\tilde{S_i} = \{j|\cos(\theta_{a_i,a_j}) > 0.2\}$. It reveals that estimating the missing content representation from the corresponding node representation doubles the accuracy in document retrieval, in comparison with the accuracy obtained by measuring the similarity based on the node representations. In particular, we are able to recover, on average, two-thirds of the similar documents in the Hepth corpus.

Table 3: Average accuracy in similar document retrieval.

	Cora	Hepth
$\tilde{S_i} = \{j\|\cos(\theta_{\tilde{b}_i,b_j}) > 0.2\}$	**0.36**	**0.65**
$\tilde{S_i} = \{j\|\cos(\theta_{a_i,a_j}) > 0.2\}$	0.13	0.30

5 CONCLUSION

In this paper, we described a way to efficiently learn a linear function to map pre-trained node and content representations learned from a network of documents. In presence of missing links and missing content, this method helps in reconstructing the neighborhood of isolated documents and also helps in recovering documents that are similar to a document whose content is unknown. In future work, we would like to investigate how to learn a non-linear transformation to refine the mapping between the two spaces.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics (TACL)* 5 (2017).
[2] Robin Brochier. 2019. Representation Learning for Recommender Systems with Application to the Scientific Literature. In *Companion Proceedings of Web Conference (WWW)*.
[3] Robin Brochier, Adrien Guille, and Julien Velcin. 2019. Global Vectors for Node Representations. In *Proceedings of The Web Conference (WWW)*.
[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.
[5] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. 2013. Information Diffusion in Online Social Networks: A Survey. *SIGMOD Rec.* 42, 2 (2013).
[6] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
[7] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*.
[8] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.
[9] Jie Liu, Zhicheng He, Lai Wei, and Yalou Huang. 2018. Content to Node: Self-Translation Network Embedding. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.
[10] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval* 3, 2 (2000).
[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).
[12] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting Similarities among Languages for Machine Translation. *CoRR* abs/1309.4168 (2013).
[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.
[15] Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31, 1 (1966).
[16] Dominic Seyler, Praveen Chandar, and Matthew Davis. 2018. An Information Retrieval Framework for Contextual Suggestion Based on Heterogeneous Information Network Embeddings. In *Proceedings of the International Conference on Research Development in Information Retrieval (SIGIR)*.
[17] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the International Conference on World Wide Web (WWW)*.
[18] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.
[19] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *Proceedings of the The Web Conference (WWW)*.