Link Prediction with Mutual Attention for Text-Attributed Networks

Robin Brochier
Université de Lyon, Lyon 2, ERIC EA3083
Digital Scientific Research Technology
robin.brochier@univ-lyon2.fr

Adrien Guille
Université de Lyon, Lyon 2, ERIC EA3083
adrien.guille@univ-lyon2.fr

Julien Velcin
Université de Lyon, Lyon 2, ERIC EA3083
julien.velcin@univ-lyon2.fr

ABSTRACT

In this extended abstract, we present an algorithm that learns a similarity measure between documents from the network topology of a structured corpus. We leverage the Scaled Dot-Product Attention, a recently proposed attention mechanism, to design a mutual attention mechanism between pairs of documents. To train its parameters, we use the network links as supervision. We provide preliminary experiment results with a citation dataset on two prediction tasks, demonstrating the capacity of our model to learn a meaningful textual similarity.

CCS CONCEPTS

• **Computing methodologies → Unsupervised learning**; *Artificial intelligence*; • **Information systems** → *Data mining*.

KEYWORDS

representation learning; link prediction; attributed network; natural language processing

ACM Reference Format:
Robin Brochier, Adrien Guille, and Julien Velcin. 2019. Link Prediction with Mutual Attention for Text-Attributed Networks. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3308560.3316587

1 RELATED WORKS

In this section, we relate recent works in the fields of network embedding (NE) and attention mechanism for natural language processing (NLP).

1.1 Attributed Network Embedding

DeepWalk [6] first proposed to derive the word embedding algorithm *Word2vec* [4] by generating paths of nodes, akin to sentences, with truncated random walks. *DeepWalk* and other variants are generalized into a common matrix factorization framework in *NetMF* [7]. To extend *DeepWalk* for text-attributed networks, *TADW* [10] expresses this latter as a matrix factorization problem and incorporates a matrix of textual features T, produced by latent semantic indexing (*LSI*), into the factorization so that the vertex similarity matrix can be reconstructed as the product of three matrices V^T, H and T.

1.2 Attention Mechanisms for NLP

The *Transformer* [9] is a novel neural architecture that outperforms state-of-the-art methods in neural machine translation (NMT) without the use of convolution nor recurrent units. The Scaled Dot-Product Attention (SDPA) is the main constituting part of the *Transformer* that actually performs attention over a set of words. It takes as input a query vector q and a set of key vectors K of dimensions d_k and value vectors V of dimensions d_v. One weight for a value is generated by a compatibility function with its corresponding key and the query. Formally, the attention vectors are generated in parallel for multiples queries Q, following the formula: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$. The result is a set of L attention vectors (L being the number of queries) of dimension d_v. The matrices Q, K and V are produced by projection of initial words representations W with three matrices P^Q, P^K and P^V whose parameters are meant to be learned.

Several works [2, 8] adapted the *Transformer* architecture beyond the task of NMT. Their main idea is to train the *Transformer* in an unsupervised fashion over large corpora of texts and further refine its parameters on a wide variety of supervised tasks. Motivated by these recent works, we present a model, *MATAN* (**M**utual **At**tention for **T**ext-**A**ttributed **N**etworks), that derives from the the SDPA to address the task of link prediction in a network of documents.

2 PROPOSED MODEL

We propose an algorithm for link prediction in text-attributed network. Our model is trained under a NE procedure, presented in Section 2.1. The optimization of the reconstruction error is performed via dot-product between contextual document representations e_u^v and e_v^u. These embeddings are generated with a mutual attention mechanism over their textual contents only, described in Section 2.2.

2.1 Overall Optimization

The model takes as input a network of documents $G = (V, E, T)$, T being the textual content of the documents. We precompute word embeddings W and a normalized similarity measure between nodes M designed from the adjacency matrix A of the network. Each document t_u is associated with a bag of word embeddings

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316587

W^{t_u} matrix. For any pair of node $(u, v) \in V^2$, mutual embeddings are generated with an asymmetric mutual attention function f_Θ^A for both documents given their bags of word embeddings $e_u^v = f_\Theta^A(W^{t_u}, W^{t_v})$ and $e_v^u = f_\Theta^A(W^{t_v}, W^{t_u})$. We define the unormalized similarity between the two nodes as the dot product of their mutual embeddings $e_u^v.e_v^u$. We aim at learning the parameters Θ by minimizing the KL divergence from the similarity distributions M (from the graph) to that of the normalized distribution of the dot products between the mutual embeddings (text associated to the nodes). We achieve this by employing noise-contrastive estimation, minimizing the following objective function: $J = -\sum_{(u,v)\in C}\Big(\log\sigma(e_u^v \cdot e_v^u) + \sum_{i=1}^k \mathbb{E}_{z\sim q}\Big[\log\sigma(-e_u^z \cdot e_z^u)\Big]\Big)$, where σ is the sigmoid function. C is a corpus of pairs of nodes generated by drawing uniformly existing links from the empirical distribution of links M. k negative nodes are uniformly drawn for each positive pair. To minimize this objective, we employ stochastic gradient descent using ADAM [3].

2.2 Mutual Attention Mechanism

The role of $f_\Theta^A(W^{t_u}, W^{t_v})$ is to generate a contextual representation of t_u given t_v. The parameters Θ we aim to learn are composed of three matrices $\Theta = \{P^Q, P^K, P^V\}$ of dimension $D\times D$ each. For all words of the target document t_u, we create queries $Q_u = W^{t_u}P^Q$. We similarly create keys and values from the contextual document t_v, such that $K_v = W^{t_v}P^Q$ and $V_v = W^{t_v}P^V$. Attention representations for each target word are then computed, following the SDPA formula: $\text{SDPA}_\Theta(W^{t_u}, W^{t_v}) = \text{softmax}(\frac{Q_u K_v^T}{\sqrt{D}})V_v$. Note that $\text{SDPA}_\Theta(W^{t_u}, W^{t_v})$ has dimension $L \times D$, that is, we have a mutual attention representation of each word of document t_u given t_v. Finally, the representation for document t_u is obtained by averaging its word mutual attention vectors: $e_u^v = f_\Theta^A(W^{t_u}, W^{t_v}) = \sum_{i=0}^L \text{SDPA}_\Theta(W^{t_u}, W^{t_v})_i$. Similarly, e_v^u is generated by flipping indices u and v. The intuition behind this model is that the matrices P^Q and P^K learn to project pairs of words that explain links in the network such that their dot-products produce large weights. P^V is then meant to project the word vectors such that their average produces similar representations for nodes that are close in the network and dissimilar for nodes that are far in the network.

3 EXPERIMENTS

To assess the quality of our model, we perform two tasks of link prediction on a dataset of citation links between scientific abstracts: Cora [1]. The first prediction evaluation, called edges-hidden, consists in hiding a percentage of the links given a network of documents and measuring the ability of the model to predict higher scores to hidden links than to non-existing ones by computing the ROC AUC. The second evaluation, called nodes-hidden, consists in splitting the network into two unconnected networks, keeping a percentage of the nodes in the training network.

We precompute on the full corpus word embeddings using *GloVe* [5] of dimension 256 with a co-occurrence threshold $x_{\max} = 10$, a window size $w = 5$ and 50 epochs. We precompute *LSI* [1] vectors of dimension 128. For the edge-hidden prediction task, we provide results performed by *NetMF* with $k = 10$ negative samples. *TADW* is run with 20 epochs and *MATAN* is performed with $k = 1$ negative sample and 10^5 sampled pairs of documents. The empirical similarity matrix between the nodes we chose is the normalized adjacency matrix. All produced representations are of dimension 256.

[1]Get the data: https://linqs.soe.ucsc.edu/data

3.1 Results

Table 1: Edges-hidden link prediction ROC AUC

% of training data	10%	20%	30%	40%	50%
NeMF	59.0	67.2	77.5	83.2	87.2
TADW	68.0	82.0	87.1	**93.2**	**94.5**
MATAN	**82.3**	**87.1**	**88.6**	90.9	91.0

Table 2: Nodes-hidden link prediction ROC AUC

% of training data	10%	20%	30%	40%	50%
TADW	64.2	**75.8**	**80.3**	**81.9**	**82.3**
MATAN	**69.4**	73.0	75.4	77.9	78.6

Tables 1 and 2 show the results of our experiments. *MATAN* shows promising results for learning on a small percentage of training data on both evaluations. *TADW* has better scores for nodes-hidden predictions which might be explained by the capacity of *LSI* to learn discriminant features on a small dataset unlike *GloVe*. In future work we would like to deal with bigger datasets from which word embedding methods might capture richer semantic information.

REFERENCES

[1] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41, 6 (1990), 391–407.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[6] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[7] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 459–467.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf* (2018).

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[10] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information.. In *IJCAI*. 2111–2117.