Global Vectors for Node Representations

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

Most network embedding algorithms consist in measuring co-occurrences of nodes via random walks then learning the embeddings using Skip-Gram with Negative Sampling. While it has proven to be a relevant choice, there are alternatives, such as GloVe, which has not been investigated yet for network embedding. Even though SGNS better handles non co-occurrence than GloVe, it has a worse time-complexity. In this paper, we propose a matrix factorization approach for network embedding, inspired by GloVe, that better handles non co-occurrence with a competitive time-complexity. We also show how to extend this model to deal with networks where nodes are documents, by simultaneously learning word, node and document representations. Quantitative evaluations show that our model achieves state-of-the-art performance, while not being so sensitive to the choice of hyper-parameters. Qualitatively speaking, we show how our model helps exploring a network of documents by generating complementary network-oriented and content-oriented keywords.

CCS CONCEPTS

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

KEYWORDS

Representation learning; network embedding; matrix factorization

ACM Reference Format:
Robin Brochier, Adrien Guille, and Julien Velcin. 2019. Global Vectors for Node Representations. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313595

1 INTRODUCTION

Networks are ubiquitous. The Web is a large-scale network of resources, social media help developing broad online social networks [6], the scientific literature forms a vast network of documents, from which one can derive a network of co-authors [16], *etc.* Understanding and exploring these networks involves solving tasks like node classification or link prediction. Efficiently solving these tasks via machine learning requires meaningful representations of the nodes.

The usual approach is to learn node representations using techniques originally devised for word embedding, based on the distributional hypothesis [13]. The analogy between word embedding and network embedding makes sense, because of the similarities in some of the statistical properties of networks and language [11]. DeepWalk [11], arguably the most popular network embedding algorithm, consists in extracting sequences of nodes, akin to sentences, via truncated random walks and then learning node representations based on the Skip-Gram model [9], using the hierarchical softmax approximation. Node2Vec [5] builds on DeepWalk and suggests another strategy for extracting node sequences via biased truncated random walks. It learns the node representations based on the Skip-Gram model, using the negative sampling approximation (SGNS). Metapath2vec [3] adapts DeepWalk to heterogeneous networks. Other variants based on SGNS are generalized into a common frame in [12].

Dealing with large-scale networks requires low-complexity algorithms. An issue with these algorithms is that SGNS scales linearly in the size of the corpus of node sequences. GloVe [10], the main alternative of SGNS, hasn't been investigated yet for network embedding, even though it scales sub-linearly in the size of the corpus. Still, GloVe is limited in the sense that it ignores non co-occurrence, as opposed to SGNS, which could result in less relevant node representations.

In this paper, we address these two issues and propose a matrix factorization approach for network embedding, inspired by GloVe. Our contributions are the following:

- we present a general model for network embedding, that consists in factorizing a thresholded co-occurrence matrix. By formulating a regression problem on all positive entries and randomly sampled zero entries, this model can take both co-occurrence and non co-occurrence into account, while preserving a competitive time complexity;
- we show how to extend this general model to networks of documents, by jointly learning word, document and node representations;
- we quantitatively assess the performance of the general model on well-known networks, against recent baselines. Not only we show that it outperforms GloVe by a very large margin on several datasets, but we also show that it outperforms on-par with recent network embedding algorithms;
- we quantitatively and qualitatively show that our model brings interesting innovation to deal with network of short

[*] These authors contributed equally to the work.

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

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313595

documents. We show how to leverage the extension of our model to explore such networks by suggesting keywords.

The rest of the paper is organized as follows. In Section 2 we survey related work. We present in details our general model, discuss its relationship to other models, and show how to deal with networks of short documents by incorporating text into the model in Section 3. Next, in Section 4, we present a thorough experimental study, where we assess the performance of our model following the usual evaluation protocol on a node classification task for well-known networks. We also discuss hyper-parameter sensitivity, and evaluate, quantitatively speaking, the extended model for networks of documents. In Section 5, we present a case study that illustrates the recommendation of keywords with the extended model. Lastly, we conclude this paper and provide future directions in Section 6.

The code for both our model and the evaluation procedure are made publicly available[1].

2 RELATED WORK

The quality and informativeness of data representation greatly influence the performance of machine learning algorithms. For this reason, a lot of efforts are devoted to devising new ways of learning representations [1]. Word embedding, *i.e.*, the task of learning representations of words, is tightly connected to the task of learning representations of nodes, *i.e.* network embedding. In this section, we first cover important works related to word embedding and then survey recent developments in network embedding.

2.1 Word Embedding

The distributional hypothesis is the basis for word embedding. It states that distributional similarity and meaning similarity are correlated, which allows learning representation of words based on the contexts in which they occur, the context of a word being co-occurring words [13]. Co-occurrences are observed by sliding a window over a large corpus.

The Skip-Gram [9] model learns word representations by maximizing the log-likelihood of a multiset of co-occurring word pairs, C:

$$\sum_{(w_i,w_j)\in C} \log p(w_j|w_i). \tag{1}$$

The conditional probability is given by the softmax function, parameterized by the word vectors:

$$p(w_j|w_i) = \frac{e^{u_i \cdot v_j}}{\sum_W e^{u_i \cdot v_w}}. \tag{2}$$

This formulation is impractical because of the cost of computing the denominator. For this reason, two variants are introduced in [9]. Skip-Gram with Hierarchical Softmax (SGHS) uses a binary tree to approximate $p(w_j|w_i)$ and speed-up learning. Skip-gram with Negative Sampling (SGNS) redefines $p(w_j|w_i)$ to make it easier to compute:

$$p(w_j|w_i) = \frac{1}{1+e^{-u_i \cdot v_j}} = \sigma(u_j \cdot v_j). \tag{3}$$

The objective becomes maximizing the following log-likelihood:

$$\sum_{(w_i,w_j)\in C} \Big(\log \sigma(w_j \cdot w_i) + \sum_{k=1}^{K} \mathbb{E}_{w_k \sim q(w_k)} \big[\log \sigma(-w_k \cdot w_i) \big] \Big). \tag{4}$$

It boils down to a classification task that consists in distinguishing the pairs of co-occurring words in C from random pairs of words, *i.e.* the negative samples. For each $(w_i, w_j) \in C$, K negative samples (w_i, w_k) are drawn, with $q(w_k) \propto \text{frequency}(w_k)^{\frac{3}{4}}$.

The GloVe [10] model learns word representations by factorizing the word-word co-occurrence matrix. Its objective is minimizing the reconstruction error, only for positive entries of X:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} f(x_{ij}) \big(u_i \cdot v_j + b_i^U + b_j^V - \log(x_{ij}) \big)^2, \tag{5}$$

where $f(x_{ij})$ is the following weighting function, that notably reduces the importance of rare co-occurrences and filter-out zero entries:

$$f(x_{ij}) = \begin{cases} \big(x/x_{\max}\big)^{\frac{3}{4}} & \text{if } x < x_{\max}, \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

The authors show that the distribution of x_{ij} follows a power-law and that the time complexity of GloVe is $O(|C|^{\frac{1}{\alpha}})$, α being the exponent of the power-law. Because α is usually larger than 1 for text, $\frac{1}{\alpha}$ becomes smaller than 1. Hence, this model has a better time complexity than Skip-gram with Negative Sampling, which runs in $O(|C|)$.

In [8], Levy and Goldberg note that the introduction of a bias for each target/context word adds an extra degree of freedom to the GloVe model as compared to the Skip-gram model.

2.2 Network Embedding

Even though the distributional hypothesis originated in linguistics and is naturally leveraged for word embedding, Perozzi *et al.* establish the connection with network embedding. To do so, 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 [11].

They propose DeepWalk, that consists in applying skip-gram with hierarchical softmax on a corpus of node sequences, deemed equivalent to sentences, generated with truncated random walks [11]. For some specific tasks, the representations learned with DeepWalk offer large performance improvements. Thus, many subsequent works focus on modifying or extending DeepWalk. Node2vec replaces random walks with 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 [5]. Dong *et al.* address the heterogeneous network representation learning problem. They propose Metapath2vec [3], a modification of DeepWalk based on meta-path-based random walks to generate sequences of heterogeneous nodes. They also suggest learning the representation with negative sampling instead of hierarchical softmax. In [18], Yang *et al.* prove that skip-gram with hierarchical softmax can be equivalently formulated as a matrix factorization problem. They then propose Text-Associated DeepWalk (TADW), to deal with networks of documents. TADW consist

[1] https://github.com/brochier/gvnr

Table 1: Notations.

Notation	Definition
n	Number of nodes.
d	Embedding dimension.
C	Corpus of co-occurring nodes.
$U \in \mathbb{R}^{n \times d}$	Target node embeddings.
$V \in \mathbb{R}^{n \times d}$	Context node embeddings.
$l \in \mathbb{N}_*^+$	Window size for observing co-occurrence.
$X \in \mathbb{R}^{n \times n}$	Co-occurrence matrix.
n_i	Number of nodes that co-occur with node i.

in constraining the factorization problem, with a pre-computed representation of documents via LSA [2].

Qiu *et al.* [12] provide the theoretical connections between Skip-Gram based network embedding algorithms and the theory of graph Laplacian. This allows them to unify DeepWalk, LINE [15] (which they prove to be a special case of DeepWalk), PTE and Node2Vec, all with with negative sampling, into the matrix factorization framework.

3 MODEL FORMULATION

In this section, we present our model, *GVNR* (Global Vectors for Node Representation), to learn node representations, taking into account both co-occurrence and non co-occurrence. We formulate a factorization problem on the thresholded co-occurrence matrix. More specifically, we formulate this problem so that the reconstruction error is measured on all the positive entries and some randomly sampled zero entries. We begin by listing the set of notations we use in Table 1, next we describe the matrix to factorize and then formulate the factorization problem. Eventually, we discuss the relationship to other models.

3.1 Description of the Matrix to Factorize

We observed node co-occurrences in truncated random walks [11]. Then, for each node j visited within q steps, with $q \leq l$, from a node i, we increase X_{ij} by $\frac{1}{q}$ [10]. Thus, we construct a weighted co-occurrence matrix, so that distant co-occurrences are increasingly downweighted. The matrix obtained with this procedure is approximately proportional to the weighted sum of the l first of the adjacency matrix: $\sum_{i=1}^{l} \frac{1}{i} A^i$. However, this sum is likely to give a denser co-occurrence matrix because unlikely and distant co-occurrences will lead to coefficients close to zero. On the contrary, the truncated random walk is likely to estimate these coefficients as exactly zero. Because we consider coefficients close to zero as noise, we're not interested in calculating them, and can computationally benefit from a sparser matrix. For the same reason, we zero-out coefficients that are less than a threshold x_{min}, assuming they are irrelevant.

3.2 Formulation of the Factorization Problem

We formulate a factorization problem on X, measuring the error only for positive coefficients and a fraction of randomly sampled zero coefficients. Note that we measure the error w.r.t the logarithm, which help compress the range of values in X [10]. Because the matrix is already thresholded, we assume all the remaining positive entries have the same importance, thus we don't weight the least-square objective:

$$\underset{U,V,b^U,b^V}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} s(x_{ij}) \Big(u_i \cdot v_j + b_i^U + b_j^V - \log(c + x_{ij}) \Big)^2. \quad (7)$$

The constant $c \in]0;1]$ allows for smoothing X while making the logarithm negative when $x_{ij} = 0$. The function s effectively selects the coefficients considered for measuring the reconstruction error:

$$s(x_{ij}) = \begin{cases} 1 & \text{if } x_{ij} > 0, \\ m_i & \text{else, with } m_i \sim \text{Bernoulli}(\alpha_i). \end{cases} \quad (8)$$

It takes the value 1 for all positive coefficients of X, while for zero coefficients, its value is given by a Bernoulli random variable, m_i, with a node-specific parameter α_i. Denoting the proportion of positive coefficient on the i$^{\text{th}}$ row of X by p_i, α_i is calculated in terms of the odd-ratio:

$$\alpha_i = \begin{cases} k \times \frac{p_i}{1-p_i} & \text{if } p_i \leq (k+1)^{-1}, \\ 1 & \text{else} \end{cases} \quad (9)$$

where $k > 0$ is an hyper-parameter that controls the proportion of zero coefficients incorporated into the calculation of the reconstruction error, akin to the number of negative samples in SGNS. The larger k, the more importance is given to pushing away vectors of non co-occurring nodes.

3.3 Relationship to Other Models

This objective function bears a resemblance to the objective of GloVe, still the two are quite different. As stated by the equation 5, GloVe performs a weighted least-square regression on the raw co-occurrence matrix. Although the authors of GloVe claim that rare co-occurrences are noisy and carry less information than the more frequent ones, this objective function still takes them into account (with a proportionally smaller weight, according to the equation 6). Our model has a stronger interpretation of this claim, by zeroing out rare co-occurrences, and weighting equally all the others. In addition, while all zero coefficients are ignored in GloVe due to the definition of f in equation 6, we incorporate a fraction of them, proportional to the quantity of positive coefficients. That constitutes an additional set of constraints we think should lead to better representations. Lastly, thresholding X helps us eliminating noise while drastically sparsifying it, since x_{ij} follows a power law [10, 11].

The complexity of Skip-Gram based algorithms, implemented with the procedure described in [9], is linear in the size of the multiset of pairs of co-occurring nodes, *i.e.* $O(|C|)$. Based on the proof given in [10], the complexity of our model is $O(|C|^{\frac{1}{\alpha}})$, where α is the exponent of the power law that models x_{ij}. For the networks studied in this paper, we observe a mean value for $\frac{1}{\alpha}$ of 0.79.

3.4 Extension to Networks of Documents

Lastly, we show how to extend the general model under the name *GVNR-t*, to deal with networks where nodes are text documents.

Assuming word order is negligible for documents [2], we can model a text as a bag of words and thus represent it by a vector $\delta \in \mathbb{N}^{+m}$, m being the size of the vocabulary. We can further

Table 2: General properties of the studied networks.

	$\|V\|$	$\|E\|$	# labels	weighted	multi-label
Citation 1	2,708	10,556	7	no	no
Citation 2	3,312	9,226	6	no	no
Co-authorship	5,021	29,856	5	yes	no
Protein	3,890	76,584	50	no	yes
Language	4,777	184,812	40	yes	yes

assume that the meaning of a text can be captured by averaging the representations of its words [7]. Therefore, with $W \in \mathbb{R}^{m \times d}$ a word embedding matrix, rather than learning the the context-vector of a node as explained previously, we define it as the average of the representations of the words it contains:

$$\underset{U,V,b^U,b^V}{\text{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} s(x_{ij}) \Big(u_i \cdot \frac{\delta_j \, W}{|\delta_j|_1} + b_i^U + b_j^V - \log(c + x_{ij}) \Big)^2, \quad (10)$$

where δ_j is the bag of words representation of text associated to the node j, and $|\delta_j|_1$ is the number of words in it. Thus, the model jointly learns node and word representations, that in turn allows representing documents.

4 QUANTITATIVE EVALUATION

A common task in network analysis is node classification. Following the experimental designs in recent works [12, 18], we assess the quality of the representations learned with *GVNR* by using them as input of a linear classifier to solve multi-class and multi-label classification tasks.

4.1 Networks

To show the versatility of *GVNR*, we consider five networks of various nature:

- Two citations networks: Citation (1) and Citation (2), extracted respectively from Cora and Citeseer[2]; each node is an article and is labelled with a conference.
- A co-authorship network extracted from DBLP[3]; each node is an author labelled with a domain of expertise[4] and each edge is weighted according to the number of common publications.
- A protein-protein interaction (PPI) network [14] which is a subgraph of the PPI network for Homo Sapiens. Each node is associated to several labels that represent biological states.
- A language network that describes collocated words observed in the first 10^8 bytes of the English Wikipedia[5] (as of March, 2006); each node is associated to multiple labels, which are the potential part-of-speech tags identified with the Stanford POS tagger [17].

The general properties of these five networks are reported in Table 2.

4.2 Tasks and Evaluation Metrics

For each network, we consider a classification task and evaluate the performance of a linear classifier, namely a logistic regression, using node representations as input. We use the LIBLINEAR implementation [4], without regularization, for a fair comparison between different representations. For each network, we train and evaluate the classifier by cross-validation, with varying split ratios.

More precisely, we consider a multi-class classification problem for the citation and co-authorship networks and measure the overall accuracy of the one-vs-all logistic regression. We consider a multi-label classification problem for the protein-protein and the language networks and measure the F_1 score of the one-vs-all logistic regression, following the procedure described in [12].

4.3 Compared Representations

We run $\gamma = 80$ random walks per node of length $t = 40$ and apply a sliding window of size $l = 5$ to generate a multiset of co-occurring nodes and then learn representation with $d = 80$ with four algorithms:

- DeepWalk: we report the results with the hierarchical softmax approximation.
- GloVe: we report the results with $x_{\text{max}} = 10$.
- NetMF: we report the results with the small-scale implementation provided by the authors, with $k = 10$ negative samples.
- *GVNR*: we report the results with $c = 1$, $x_{\text{min}} = 1$ and $k = 1$ and discuss their impact in the next sub-section.

4.4 Result Analysis

Tables 3 to 7 detail the accuracy measures. The classifier performs well with the representations learned by *GVNR*, achieving similar or better results w.r.t the representations learned with DeepWalk and NetMF. Still, there is one exception concerning the language network, where the classifier performs best with the representation learned with GloVe. We sum up our findings in two points:

(1) *GVNR* always achieves a good performance, while there is more variance in the performance of the baselines. For instance, NetMF is largely outperformed by the others on the language network, GloVe is also significantly outperformed on the co-authorship network, while DeepWalk struggles to learn good representations from the citation (2) network.
(2) Thresholding always improves the performance of *GVNR*. As an example, its leads to an average gain of 9.4 accuracy on the co-authorship network.

Interestingly, GloVe produces the best results on the language network. The average number of neighbors is 38.7 for this network which makes it the denser network.

4.5 Impact of the Hyper-Parameters

Figure 1 shows the sensitivity, in terms of accuracy, of *GVNR* to its hyper-parameters, namely, the threshold x_{min}, the shifting constant c, the window length l, and the sampling proportion k. We only report the results for the citation (1) network since their are similar across the considered networks. We see that c has little impact on the accuracy. In practice, we found *GVNR* performs best across the

[2] https://linqs.soe.ucsc.edu/data
[3] https://dblp.uni-trier.de/
[4] https://static.aminer.org/lab-datasets/expertfinding/
[5] http://mattmahoney.net/dc/text.html

Table 3: Accuracy on the citation (1) network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	57.7	62.4	69.5	72.8	73.8
GVNR ($x_{\min} = 0$)	58.5	62.5	70.7	73.4	75.0
NetMF	65.7	**72.9**	**76.4**	**78.6**	79.4
DeepWalk	67.8	71.6	74.5	75.8	79.2
GVNR ($x_{\min} = 1$)	**69.5**	72.6	75.9	78.1	**80.2**

Table 4: Accuracy on the citation (2) network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	42.8	53.5	55.3	56.2	56.8
GVNR ($x_{\min} = 0$)	38.7	46.8	49.1	50.4	50.9
NetMF	**51.2**	54.8	55.1	55.0	54.8
DeepWalk	41.3	52.5	54.5	55.5	56.0
GVNR ($x_{\min} = 1$)	45.6	**55.6**	**57.3**	**58.7**	**59.0**

Table 5: Accuracy on the co-authorship network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	41.0	42.1	43.7	46.4	51.2
GVNR ($x_{\min} = 0$)	60.3	64.6	67.4	67.1	68.2
NetMF	60.7	66.2	70.1	72.1	72.8
DeepWalk	**75.4**	**77.2**	**77.3**	**75.9**	**79.3**
GVNR ($x_{\min} = 1$)	74.7	75.3	76.3	73.8	74.6

Table 6: F_1 score on the language network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	**34.0**	**44.1**	**46.7**	**47.7**	**48.6**
GVNR ($x_{\min} = 0$)	31.7	40.7	43.2	44.7	45.1
NetMF	27.5	33.5	36.2	37.7	38.7
DeepWalk	33.6	43.6	46.2	47.6	48.2
GVNR ($x_{\min} = 1$)	32.2	41.7	44.0	45.2	46.1

datasets with $c = 1$. It seems that the model performs best with a threshold value between 1 and 5. Setting $x_{\min} > 0$ clearly improves the performance. In general, we observed that $k = 1$ constantly brings an improvements over $k = 0$, but higher values only brings a boost for high training ratios. Finally, the accuracy increases along with the window size l, with few improvements above $l = 5$.

4.6 Additional Results with Text

We now report additional results when taking into account the text information, associated to the nodes of the citation networks, *i.e.* titles and abstracts. We consider three ways of learning text-aware representations: TADW [18], the representations learned with DeepWalk concatenated with the LSA representation of the texts, and *GVNR-t*. We report the results for TADW with 20 iterations and 4 iterations for *GVNR-t*.

Table 7: F_1 score on the protein-protein network.

	% of training data				
	10%	20%	30%	40%	50%
GloVe	11.8	13.8	15.6	17.3	19.1
GVNR ($x_{\min} = 0$)	10.7	13.3	15.0	16.7	18.1
NetMF	10.2	11.7	13.5	14.7	16.1
DeepWalk	**12.2**	**13.9**	**16.2**	**17.8**	**19.6**
GVNR ($x_{\min} = 1$)	11.7	13.3	15.8	17.6	19.2

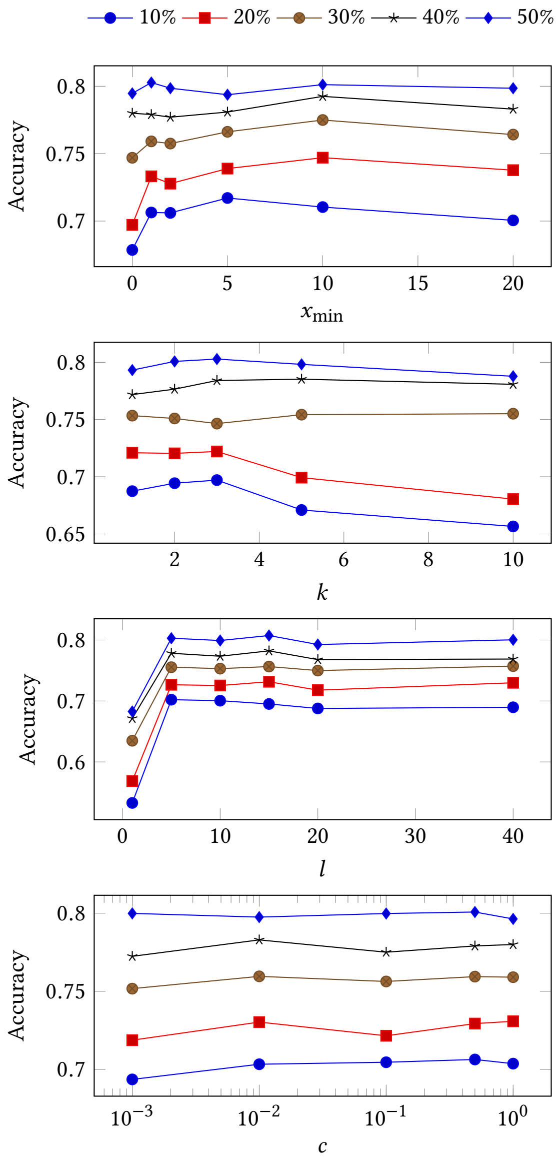

Figure 1: Sensitivity of *GVNR* to the hyper-parameters $x_{\min}$, k, l and c on the citation (1) network.

Tables 8 and 9 report the accuracies. *GVNR-t* shows interesting improvements over the simple concatenation of text and graph features and the reference baseline in the field, TADW. This motivates the study of the word representations learned by *GVNR-t* and the interplay between the node and document representations.

Table 8: Accuracy on the citation (1) network, considering the text features.

	% of training data				
	10%	20%	30%	40%	50%
LSA	54.7	61.0	62.4	63.0	62.8
DeepWalk+LSA	73.8	77.9	78.4	78.1	78.1
TADW	77.1	78.8	78.2	78.8	78.6
GVNR-t	**79.3**	**80.7**	**80.8**	**81.4**	**81.1**

Table 9: Accuracy on the citation (2) network, considering the text features.

	% of training data				
	10%	20%	30%	40%	50%
LSA	52.0	54.7	54.7	58.4	65.7
DeepWalk+LSA	58.3	60.7	61.1	60.0	61.2
TADW	60.6	60.1	60.1	66.2	69.3
GVNR-t	**63.3**	**62.5**	**64.9**	**68.6**	**70.4**

5 CASE STUDY

To showcase the usefulness of learning jointly word, node and document representation, we apply *GVNR-t* to full DBLP network that consists in 1,397,240 documents and 3,021,489 citation relationships. After computing X, we threshold it with $x_{\min} = 20$, which divides the density of $X_{(ij)}$ by 50. For the learning phase, we keep the same settings as before, that is $d = 80$, $k = 1$, $c = 1$, $l = 5$. Note that we apply a standard pre-processing, that consists in merging recurrent phrases, as suggested in [9]. Computing X and estimating U and W, the node and word representations takes about 8 hours on a single machine with 32 cores and 192 GB of RAM.

Our aim is to show that we can measure the similarity, on the one hand, between word and node representations, and, on the other hand, between word and document representations, to extract sets of complementary keywords.

Tables 10 and 11 show 2 randomly selected papers for which we computed the 5 closest word embeddings w_k to, respectively, (i) the node representation u and (ii) its content representation v. First, we note that the keywords are all relevant, even though none of them actually appear in the documents. Then, we see that words close to the node representation are more general, giving a broad view of the studied topic, whereas words close to the document representation are more specialized and provide a more fine-grained perception. We suspect that the network topology, mostly influencing U, helps locating a node in its broader context, while the content of documents, mostly influencing W, helps in selecting some very specific keywords. In future work, we would like to investigate further the quality of the learned representations for a wider range of recommendation tasks.

Table 10: Keyword recommendation by selecting the closest word embeddings w_k to both embeddings u (node) and v (content) of an input document (1).

Document	**A brief survey of computational approaches in social computing** Web 2.0 technologies have brought new ways of connecting people in social networks for collaboration in various on-line communities. Social Computing is a novel and emerging computing paradigm...
Closest words to u (node)	*cold start problem, storylines, document titles, movielens data, computational humor*
Closest words to v (content)	*social, social network, enron email corpus, social networks, extremely large datasets, sites blogs*

Table 11: Keyword recommendation by selecting the closest word embeddings w_k to both embeddings u (node) and v (content) of an input document (2).

Document	**Discovering company revenue relations from news** A network approach Large volumes of online business news provide an opportunity to explore various aspects of companies. A news story pertaining to a company often cites other companies. Using such company citations we...
Closest words to u (node)	*datenbanksystemen, denizens, want hear, technological infrastructures, asynchronous discussions*
Closest words to v (content)	*company, consumer brand, today highly competitive, data cleaning, consumer heterogeneity*

6 CONCLUSION

In this paper, we presented *GVNR*, a matrix factorization based method for network embedding, that better handles non co-occurrence than GloVe. We further extended this model to incorporate textual content associated with the nodes to learn meaningful representations of words and documents. We showed that *GVNR* performs state-of-the-art results on a wide range of networks and can provide recommendations based on the distance between words, documents and graph embeddings. In future works we would like to explore further the relations between the word embeddings learned with *GVNR-t* and traditional word embeddings learned with GloVe or Skip-Gram. More particularly, we would like to study whether *GVNR-t* could help learning better word embeddings from small structured corpora.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[2] 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.

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[4] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research* 9, Aug (2008), 1871–1874.

[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[6] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. 2013. Information Diffusion in Online Social Networks: A Survey. *SIGMOD Record* 42, 2 (2013), 17–28.

[7] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*. 1188–1196.

[8] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3 (2015), 211–225.

[9] 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*. 3111–3119.

[10] 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.

[11] 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.

[12] 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.

[13] Magnus Sahlgren. 2008. The distributional hypothesis. *Italian journal of linguistics* (2008), 23–53.

[14] Chris Stark, Bobby-Joe Breitkreutz, Andrew Chatr-Aryamontri, Lorrie Boucher, Rose Oughtred, Michael S Livstone, Julie Nixon, Kimberly Van Auken, Xiaodong Wang, Xiaoqi Shi, et al. 2010. The BioGRID interaction database: 2011 update. *Nucleic acids research* 39, suppl_1 (2010), D698–D704.

[15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. 1067–1077.

[16] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 990–998.

[17] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 173–180.

[18] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2111–2117.