Word-Node2Vec: Improving Word Embedding with Document-Level Non-Local Word Co-occurrences

Procheta Sen[1]
procheta.sen2@mail.dcu.ie

Debasis Ganguly[2]
debasga1@ie.ibm.com

Gareth J.F. Jones[1]
Gareth.Jones@dcu.ie

[1] ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
[2] IBM Research, Dublin, Ireland

Abstract

Standard word embedding algorithms, such as word2vec and Glove, make a restrictive assumption that words are likely to be semantically related only if they co-occur locally within a window of fixed size. However, this restrictive assumption may not capture the semantic association between words that co-occur frequently but non-locally within documents. To alleviate this restriction, in this paper, we propose a graph-based word embedding method, named 'word-node2vec'. By relaxing the strong constraint of locality, our method is able to capture both local and non-local co-occurrences. Word-node2vec constructs a weighted graph, where each node represents a word and the weight of an edge between two nodes represents a combination of both local (e.g. word2vec) and document-level co-occurrences. Our experiments show that word-node2vec outperforms word2vec and glove on a range of different tasks, such as word-pair similarity prediction, word analogy and concept categorization.

1 Introduction

Word embedding, the process of obtaining vector representations of words, is a first step towards addressing language semantics, in which discrete entities, such as words, are embedded as vectors over a continuous space of reals. This not only facilitates to obtain semantic similarities between words to improve tasks such as semantic search (Ganguly et al., 2015; Roy et al., 2016), but is also useful in a number of down-stream NLP tasks including concept categorization (Jastrzebski et al., 2017), information retrieval (Guo et al., 2016), sentence similarity prediction (Mueller and Thyagarajan, 2016), sentiment analysis (Faruqui et al., 2015) and POS tagging (Tsvetkov et al., 2016) etc.

Word embedding approaches such as word2vec (Mikolov et al., 2013a) and Glove (Pennington et al., 2014) rely on a large corpus to learn the association between words. The architecture of existing word embedding approaches mimics the process of human cognition of word association by learning the representation of each word with an objective of maximizing the likelihood of predicting the words around its *local* context (defined by a fixed length word window). A limitation of existing word embedding approaches, such as word2vec and glove, is that they use a strong constraint that words are likely to be semantically related to each other only if one occurs within a local context of the another, where the local context is given by a word window of specified length.

On the other hand, non-local or document-level co-occurrences between words have been widely used to estimate semantic similarities between words. More specifically, the latent semantic analysis (LSA) method proposed by Deerwester et al. (1990) uses a spectral analysis (method of principal component analysis) of the term-document matrix of a collection to obtain the most informative *concepts* (word classes), and then expresses each document as a linear combination of these principal components. Blei et al. (2003) estimate a generative model from a given collection by assuming that documents are mixtures of a preset number of topics, where each topic represents a word distribution over the vocabulary. This is largely similar to decomposing a term-document matrix as a product of matrices with non-negative components, a process commonly known as non-negative matrix factorization (NMF) (Gaussier and Goutte, 2005). The underlying common idea among all these approaches is to make use of the frequent document-level word co-occurrences to identify likely semantic association between words.

Despite the presence of a vast volume of lit-

Proceedings of NAACL-HLT 2019, pages 1041–1051
Minneapolis, Minnesota, June 2 - June 7, 2019.

erature on document-level (non-local) word co-occurrences, word embedding approaches do not utilize this information to derive the word representations. In this paper, we propose to augment the document-level non-local word co-occurrence information with the local co-occurrence information that methods such as word2vec and glove use. More specifically, we propose a graph-based word embedding method, named `word-node2vec`, that by relaxing the strong constraint of locality, is able to capture both the local and non-local co-occurrences. To represent the local dependencies, each node, representative of a word (hence the name 'word-node'), is initialized with a vector representation obtained with a standard method, e.g. word2vec. We then define the weight of the edge between a pair of word-nodes to reflect their likelihood of non-local co-occurrence, computed with the help of the global term-document matrix for the whole collection.

The rest of the paper is organized as follows. In Section 2, we survey existing literature on word embedding. In Section 3, we revisit the skip-gram approach and propose a graph-based view of the skip-gram objective as a pre-cursor to developing our model. In Section 4, we extend the skip-gram graph model with non-local document-level co-occurrence information. Section 5 describes our experimental setup. Section 6 reports the results of our new embedding approach against a number of baselines. Finally, Section 7 concludes the paper with directions for future work.

2 Related Work

The word2vec (Mikolov et al., 2013a) embedding model shifts a window of a predefined size (a parameter) across the text of a collection of documents in order to train a linear classifier for each word to predict itself given its context (continuous bag-of-words), or its context given the word (skip-gram). The parameter vector transforming a word to its context (or vice-versa) gives its embedded representation. In addition to making use of the words in the context as positive samples, word2vec also relies on the use of words randomly sampled from the collection (outside the current context) as negative examples. Levy and Goldberg (2014) showed that the negative sampling based skip-gram (SGNS) objective function of `word2vec` is mathematically equivalent to factorizing a positive point-wise mutual information gain (PPMI) matrix shifted by $log(k)$, where k is the number of negative samples.

The key idea behind the glove algorithm proposed in (Pennington et al., 2014) is to make use of the ratio of the co-occurrence probabilities between word pairs to better distinguish semantically related words from non-related ones. The study ultimately shows that factorizing the log of the co-occurrence matrix leads to effective embedded representation of words. The co-occurrences in both word2vec and glove are essentially local in nature. In contrast, our proposed algorithm leverages both local and non-local co-occurrences.

More recently, Peters et al. (2018) proposed `ELMO`, a deep contextualized word representation with layers of stacked bi-directional LSTMs to model both a) complex characteristics of word use (e.g., syntax and semantics), and b) their diversity across various linguistic contexts. A limitation of ELMO is that a word representation may effectively be learned mainly in the presence of an associated context, as a result of which the method is likely to find applications mostly in downstream tasks, e.g. question answering and sentiment analysis. However, in contrast, our proposed method can learn the representation of a word in isolation, which means that, similar to word2vec and Glove, word vectors obtained using our method can be applied directly to (and is also likely to work well for) word similarity and word analogy tasks. We included ELMO as of our baseline approaches in our experiments.

Grover and Leskovec (2016) proposed a skip-gram based objective function to embed each node of a graph. Analogous to skip-gram based word embedding, each node vector is given as input to a linear classifier to predict the context vector around a node. The context vector around a node, in this case, consists of a sequence of nodes visited by a random walk starting from that node. In our method, we use a similar graph-based construction to train vector representations of a node (each node a word). However, we use a stratified sampling approach within a maximum distance (hop-count) of 2, instead of allowing the random walk to proceed along in a combined depth-first and breadth-first manner, as in (Grover and Leskovec, 2016). Through our experiments, we find that larger hop-counts (i.e. longer transitive dependencies) introduce noise in the document-level word co-occurrence estimation process.

3 Generalized Word Embedding

In this section, we propose a general word embedding framework based on the skip-gram objective function of word2vec. Our proposed method relies on a general construction of the *context* around a word. We modify the skip-gram objective function of word2vec to take into account this general context of words. Before describing our proposed approach, we revisit the objective function of negative sampling based skip-gram word2vec (SGNS).

Skip-gram. In word2vec, the context of a word comprises words occurring within a window of a fixed size (say k) pivoted at a particular instance of w in the collection. More formally, let $\Lambda(w)$ denote the set of indexes where the word w occurs in a collection $C = \{t_1, \ldots, t_T\}$, T denoting the total number of tokens in the collection C, i.e.

$$\Lambda(w) = \{i : t_i = w\}. \tag{1}$$

We then construct the context $c(w)$ of a word as

$$c(w) = \cup_{i \in \Lambda(w)} \cup_{\substack{j=-k \\ j \neq 0}}^{k} t_{i+j} \tag{2}$$

Let Ω denote the set of all observed word-context pairs $(w, c(w))$, i.e.

$$\Omega^+ = \cup_{w \in V} \{w, c(w)\}, \tag{3}$$

where V denotes the vocabulary set, and Ω^- denote the set of negative samples of word-context pairs, i.e.

$$\Omega^- = \cup_{w \in V} \{w, \cup \{v : v \sim (V - c(w))\}\}, \tag{4}$$

where words v's in the negative context set are randomly sampled from the complement set $c(w)$.

Let y be an indicator random variable denoting semantic relatedness of a word with its context. For a word w and its context $c(w)$ (as defined in Equation 2, the SGNS algorithm seeks to maximize the objective function

$$\begin{aligned} J(\theta) = & \sum_{w, c(w) \in \Omega^+} p(y = 1 | \mathbf{w}, \mathbf{c_w}) + \\ & \sum_{w, c(w) \in \Omega^-} p(y = 0 | \mathbf{w}, \mathbf{c_w})), \end{aligned} \tag{5}$$

where $p(.)$ is the log-likelihood function, and $\theta \in \mathbb{R}^{d \times |V|}$ represents the trainable matrix of parameters, each d dimensional column vector of the matrix θ denoting the vector representation of word w, i.e. $\mathbf{w} = \theta_w$. Note that the vector for a set of context words $c(w)$ is obtained by some aggregation function (sum or average) over the constituent words, i.e.

$$\mathbf{c}(\mathbf{w}) = \sum_{u \in c(w)} \mathbf{u}. \tag{6}$$

In order to optimize $J(\theta)$, the word2vec approach shifts a window of size k pivoted around a word $w = t_i$ (token positioned at offset i in the corpus), and applies stochastic gradient descent (SGD) to update the parameters for the corresponding word w and its context vector $c(w)$.

A Graph Formulation of SGNS. We now propose a general framework that allows contexts to be defined in a more general way. The solution relies on defining a graph $G = (\mathcal{V}, \mathcal{E})$, where each node corresponds to a word from the vocabulary of the given collection, i.e.

$$\mathcal{V} = \{x_w : w \in V\}. \tag{7}$$

In general, an edge $(x_u, x_v) \in \mathcal{E}$ represents a *relation* between two words u and v of weight $w(x_u, x_v) \in \mathbb{R}$. For example, in order to define the context of SGNS (Equation 2), the edge set is defined as

$$\mathcal{E} = \{(x_w, x_u) : u \in \cup_{i \in \Lambda(w)} \cup_{\substack{j=-k \\ j \neq 0}}^{k} t_{i+j}\}. \tag{8}$$

Learning the vector representations for each node of the graph G leads to learning the vector representation for each word, because there is a one-one mapping between the set of nodes $\mathcal{V}$ and the set of words V (henceforth we refer to a node of this general class of graphs, defined as per Equation 7, as a *word-node*). The objective of the embedding is to learn vector representations of nodes such that two nodes are close in the embedded space if, as per the edge relations of the graph, these nodes are within a κ-adjacency neighborhood of each other. The κ-adjacency neighborhood of a graph is the set

$$N_\kappa(x_w) = \{x_u \in \mathcal{V} : h(x_w, x_u) \leq \kappa\}, \tag{9}$$

where $h(u, v)$ denotes the hop-count or adjacency number between nodes u and v. In the general formulation, the set of $N_\kappa(x_w)$, constituting the set of nodes reachable from paths of length at most k starting at x_w, act as positive examples to learn the

embedding of node x_w. This is because these positive examples seek to make the vector representation of x_w similar to the vector representations of nodes in $N_\kappa(x_w)$. More formally,

$$\Omega^+ = \cup_{x_w \in \mathcal{V}} \{x_w, N_\kappa(x_w)\},$$
$$\Omega^- = \cup_{x_w \in \mathcal{V}} \{x_w, \cup \{x_u : u \sim \mathcal{V} - N_\kappa(x_w)\}\}. \quad (10)$$

Instead of iterating over the words in a corpus, the SGNS equivalent is then achieved by iterating over the set of nodes and maximizing the same objective function of Equation 5 using the definitions of the positive and negative example sets from Equation 10. Note that to achieve the SGNS objective the value of κ is set to 1 in the definition of Ω^+ in Equation 10, i.e. the set of context for a word-node comprises one-hop neighbours as defined by the edge relations of Equation 8.

4 Extending the Graph Model for Non-Local Co-occurrences

The graph based approach of Section 3 allows alternative ways to define the context and learn the objective function to obtain word-node representations. In this section, we describe how to augment the non-local document-level co-occurrence information in the graph-based framework.

Co-occurrence Weights. The first step to include non-local co-occurrences is to modify the edge relations of SGNS (Equation 8) to accommodate weighted document-level co-occurrences. Instead of considering the collection $C = \{t_1, \ldots, t_T\}$ as a stream of words, we consider C as a set of M documents $\{D_i\}_{i=1}^M$.

First, we make provision to include weighted edges of the form $(x_w, x_u, \omega(x_w, x_u))$ in the edge construction process of Equation 8. The weight $\omega(x_w, x_u)$ between word-nodes x_w and x_u is intended to represent a measure of association between these words.

Next, we describe how to compute the non-local co-occurrence weight between a pair of words. First, we compute the co-occurrence probability of two words w and u as

$$P(w,u) = \frac{\sum_{i=1}^M \mathbb{I}(w,u,D_i)}{\sum_{i=1}^M \mathbb{I}(w,D_i) \sum_{i=1}^M \mathbb{I}(u,D_i)}, \quad (11)$$

where the numerator denotes the total number of times that the words w and u co-occur in the collection of all documents, and the denominator denotes the number of times each occur independently. In our approach, we use a generalized form of Equation 11, where analogous to the Jelinek-Mercer smoothing method (Ponte and Croft, 1998), we take into account the informativeness of the co-occurrences by linearly combining the frequencies with the global statistics of inverse collection frequency. More specifically,

$$P_\alpha(w,u) = \alpha P(w,u) + \frac{(1-\alpha)T^2}{|\Lambda(w)||\Lambda(u)|}, \quad (12)$$

where $P(w,u)$ represents the maximum likelihood estimate computed by Equation 11 and the denominator denotes the product of the collection frequencies of the terms (as per the notation of Equation 1). It can be seen that Equation 12 allows relative weighting of the term frequency and the informativeness components.

Combination with Local Co-occurrences. The next step in our word-node2vec method is to augment the non-local co-occurrence information computed as per Equation 12 with the local co-occurrence of SGNS as defined in Equation 8. For this, analogous to (Pennington et al., 2014), we compute the probability of co-occurrence between a word pair restricted within a window of size k over the whole collection. More formally,

$$P_k(w,u) = \frac{1}{|\Lambda(w)|} \sum_{i \in \Lambda(w)} \mathbb{I}(t_{i+j} = u)_{j=-k}^k \quad (13)$$

Next, we assign weight to an edge by combining the local and non-local co-occurrence probabilities estimated from Equations 13 and 12 respectively. Formally speaking,

$$\omega(x_w, x_u) = P_\alpha(w,u) P_k(w,u). \quad (14)$$

Context with Weighted Edges. Constructing the context of a node x_w (Section 3), requires a modification aimed to take into account the edge weights while selecting the neighboring nodes of x_w. Instead of defining the context as the entire set of κ-neighborhood $N_\kappa(x_w)$ of a node x_w, we define a κ-neighbourhood of length (hop-count), l, which is a subset of l samples drawn from the overall neighbourhood.

The likelihood of sampling a node x_u from the neighbourhood set is proportional to the weight of the edge (x_w, x_u), i.e., $\omega(x_w, x_u)$. This way of

defining the context allows the algorithm to make use of the edge weights (local and non-local co-occurrences) in learning the node representations, i.e. assigning more importance to associations with higher weights in seeking to embed the current word-node close to them.

Our idea, in general, is to use stratified sampling, where each stratum corresponds to a neighbourhood of particular length. The priors assigned to the strata in increasing sequence of adjacency length form a decreasing sequence, which means that the most emphasis is put on direct co-occurrence evidence (i.e. the 1-adjacent neighborhood), than to the 2-adjacent nodes and so on.

Stratified sampling requires the strata to be mutually disjoint of each other. This means that the κ-neighbourhood of Equation 9 needs to be redefined to ensure that any node belongs to exactly one of the partitions (defined by its hop-count). To state this formally, we define the set of nodes of (*not up to*) hop-count j as

$$H_j(x_w) = \cup\{x_u : h(x_w, x_u) = j\} \tag{15}$$

The κ-neighbourhood is then defined as

$$N_\kappa(x_w) = \cup_{j=1}^{\kappa}(H_j(x_w) - \cup_{j'=1}^{j-1} H_{j'}(x_w)). \tag{16}$$

A subset of size l, comprised of stratified samples from $N_\kappa(x_w)$, is then sampled with decreasing priors $\beta_1, \ldots, \beta_\kappa$, i.e., $\beta_j < \beta_{j-1} \forall j = 2, \ldots, \kappa$ and $\sum_{j=1}^{\kappa} \beta_j = 1$.

Putting things together, the probability of sampling a node from the set $N_\kappa(x_w)$ defined as per Equation 16 is then given by

$$P(x_u|N_\kappa(x_w)) = \beta_j P(x_u|H_j(x_w)) = \beta_j \frac{\omega(x_w, x_u)}{\omega(x_w, .)}, \tag{17}$$

where $\omega(x_w, x_u)$ are edge weights computed with Equation 14 and $\omega(x_w, .)$ denotes the sum of edges emanating from node x_w.

As a point of note, for our experiments, we obtained optimal results by using $\kappa = 2$. Consequently, to simplify the description of our experiments, we name the parameter β_1 as β (the parameter β_2 is then identical to $1 - \beta$). We would also mention at this point that our proposed way of constructing the context by sampling neighboring nodes is different from the one proposed in (Grover and Leskovec, 2016), which uses a combination of breadth-first (BFS) and depth-first (DFS) traversals, with parameters p and q respectively.

#Documents	4,641,754
#Avg. Doc Length (#words)	43.23
#Vocabulary size	461,572
#Tokens	202,575,916

Table 1: Dataset characteristics of DBPedia-2014.

Our experiments reveal that our sampling strategy outperforms that of Grover and Leskovec (2016) (treated as a baseline).

5 Experimental Setup

In this section, we describe our experimental setup to evaluate our new word embedding method.

5.1 Dataset

A word embedding algorithm requires a collection to learn word representations. To compare the various word embedding approaches (i.e. our method and the baselines), we use the DBPedia (2014) corpus, which is a collection of abstracts of Wikipedia pages crawled in 2014[1]. Dataset characteristics are outlined in Table 1. As part of preprocessing, we removed words with collection frequency less than 10 and also removed stopwords[2].

5.2 Baselines and Implementation

The objective of our experiments is two-fold. First, to show that a combination of local and global approaches is likely to yield effective embedded representations of word vectors, and second that our proposed graph-based formalism is likely to work better than a trivial black-box way of combining the two sources of information.

Local Co-occurrence approaches. As approaches that use local co-occurrence information, we use three state-of-the-art embedding approaches namely skip-gram word2vec with negative sampling (SGNS) (Mikolov et al., 2013a), Glove (Pennington et al., 2014) and Fasttext (Joulin et al., 2016). All these methods rely only on co-occurrences (at the level of words for the first two and at the level of character n-grams for the last one) within a word or character n-gram window of specified length k (acting as a parameter). Fasttext learns the vector representation of each word by aggregating (vector sum) the vector representations of its constituent n-grams.

[1] http://downloads.dbpedia.org/2014/en/long_abstracts_en.ttl.bz2

[2] http://www.lextek.com/manuals/onix/stopwords2.html

Additionally, we also employ a more recent approach, namely ELMO (Peters et al., 2018), which relies on a pre-trained model (comprised of stacked bidirectional LSTMs) to infer vectors for a given context (typically a sequence of words). For our experiments,

Document-level Co-occurrence approaches. Although not an embedding approach, the LDA topic modeling algorithm outputs two matrices, namely $\theta \in \mathbb{R}^{M \times d}$ and $\phi \in \mathbb{R}^{d \times V}$, representing the document-topic and topic-words distribution respectively (Blei et al., 2003). LDA uses document-level word co-occurrences to estimate both these matrices. In principle, one can then use the ϕ matrix as a substitute for the word embedding parameter matrix of SGNS (see Equation 5). This gives d dimensional vectors for each word purely with a global co-occurrence based approach.

Although it is possible to choose other non-local co-occurrence approaches as baselines, e.g. PLSA (Hofmann, 1999) or LSA, (Deerwester et al., 1990), it was shown in (Blei et al., 2003) that LDA outperforms each of these. Consequently, we use the stronger baseline of LDA in our experiments.

Combination of Local and Non-local Co-occurrences. To empirically demonstrate the effectiveness of our proposed graph-based word-node embedding, we employ an additional baseline that is a linear combination of the word vectors obtained individually with the local and non-local approaches. More formally, the vector of each word w is given as

$$\mathbf{w} = \lambda \mathbf{w}_{\text{Local}} + (1 - \lambda)\mathbf{w}_{\text{LDA}}, \qquad (18)$$

where $\mathbf{w}_{\text{Local}}$ is the vector representation of word w obtained by a local co-occurrence baseline, i.e. SGNS and Glove, whereas $\mathbf{w}_{\text{LDA}}$ represents the vector for the word w obtained with LDA.

Additionally, we employ the node2vec approach as a baseline. In particular, we use node2vec to learn the word-node representations of the graph constructed as per Section 4. The purpose of this baseline is to show that our way of defining the contexts around word-nodes is more suitable for our task of word embedding than a general-purpose graph node embedding approach.

5.3 Evaluation Tasks and Datasets

To compare the relative performance of word-node2vec with the baselines, we use a number of datasets, each corresponding to one of the following three evaluation tasks.

Dataset	Composition	Example
MSR	Syntactic	good:better rough:X
Google	Syntactic and Semantic	Athens:Greece Berlin:X
SemEval	Syntactic and Semantic	dog:bone bird:X

Table 2: Word analogy datasets overview.

Word Similarity. A standard way to measure the effectiveness of embedded words is to measure how well the similarity between a pair of words correlates with human judgments. Two such standard datasets that we use for our experiments are the WSIM-353 (Finkelstein et al., 2014) and the MEN (Bruni et al., 2014) datasets. Both comprise a list of word pairs, with an associated human judged similarity value. This similarity value is expected to be high for semantically similar words, such as 'morning' and 'sunrise' (human assigned score of 49 out of 50), and low for semantically unrelated words, such as 'angel' and 'gasoline' (score of 1 out of 50), both examples being taken from the MEN dataset.

Word Analogy. The word analogy task consists of templates of the form "A:B as C:X", where A, B, and C are given words, whereas X is unknown. Using a vector representation of words this analogy task is solved by retrieving the vector most similar to that of $\mathbf{B} + \mathbf{C} - \mathbf{A}$. A word embedding is considered effective if it finds a greater number of correct answers (resulting in higher accuracy).

We employed three different analogy datasets, namely, the Google Analogy (Mikolov et al., 2013a), the MSR Analogy (Mikolov et al., 2013b) and the SemEval-2012 task 2 (Jurgens et al., 2012) datasets. The MSR dataset contains syntactic questions only involving morphological variations. The Google dataset on the other hand contains both syntactic and semantic questions.

Given an analogy 'A:B as C:D', the Semeval-2012 task requires prediction of the degree to which the semantic relations between A and B are similar to those between C and D. In our experiments, we treat the given entity D as unknown and seek to predict D, similar to the MSR and Google analogy datasets. Table 2 provides an overview of examples from these datasets.

Concept Categorization Task. The concept categorization task requires classifying nouns into

a concept type derived from an ontology. For this task, we employ the AP (Almuhareb and Poesio, 2005), BLESS (Baroni and Lenci, 2011) and ESSL$_{2b}$ (Marco Baroni and Lenci, 2008) datasets. The AP dataset contains 402 nouns from 21 WordNet classes, e.g., nouns such as 'ceremony', 'feast', and 'graduation' belong to the class 'Social Occasion'. The BLESS dataset, designed for the evaluation of distributional semantic models, contains 200 distinct English concrete nouns as target concepts. These nouns are categorized into 17 broad classes.

Evaluation Metrics and Pipeline. The word similarity prediction effectiveness is measured with the help of Spearman's rank correlation coefficient ρ. This measures the rank correlation (higher is better) between the list of word pairs sorted in decreasing order of inter-similarity values as predicted by a word embedding algorithm and the reference list of human judged word pairs. For the analogy and the concept categorization tasks, we report the accuracy in predicting the reference word and that of the class, respectively.

Parameters and Settings. In our experiments, for all the methods, except ELMO, we set the number of dimensions to 200. To find optimal settings for each method (except ELMO), we use the MEN dataset as a development set for tuning the parameters of each method. Each method with the optimal parameter settings is then applied for the rest of the datasets and tasks.

Since we used a pre-trained model for ELMO, the number of dimensions corresponds to the size of the output layer of the network, the value of which in the default configuration of the Python implementation[3] is 1024.

The parameters of SGNS are window size (k) and the number of negative samples (NS). For the baseline approach SGNS, we varied k from 5 to 40 in steps of 5 and found that the best results are obtained when $k = 10$ and $NS = 5$. Similarly, for Glove we chose the optimal settings by varying k within the same range of $[5, 40]$ and found that the optimal ρ for the MEN dataset is obtained for $k = 20$. We obtain the LDA results by setting the number of topics to 200 (so as to match with the dimensionality). As LDA hyper-parameters, we use settings as prescribed in (Griffiths and Steyvers, 2004), i.e., $\beta = 0.1$ and $\alpha = 0.25$ $(50/(\#topics = 200))$.

[3] https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md

Method	Spearman's ρ	
	MEN	WSIM
SGNS ($k = 10$, $NS = 5$)	0.7432	0.6977
Glove ($k = 20$)	0.7066	0.6706
FastText	0.7307	0.6518
ELMO	0.4225	0.4631
LDA	0.4933	0.4074
SGNS-LDA ($\lambda = 0.9$)	0.7367	0.6548
Node2vec ($p = 0.5, q = 0.5, l = 40$)	0.7440	0.6988
Word-node2vec ($\alpha = 0.5, \beta = 0.7, l = 20$)	**0.7491**	**0.7032**

Table 3: Word similarity prediction results.

Since we found that SGNS performed significantly better than Glove, we use SGNS vectors for the linear combination method (Equation 1), which we call SGNS-LDA from hereon. The parameter λ was varied within a range of $[0.1, 0.9]$ in steps of 0.1 ($\lambda = 0$ and $\lambda = 1$ degenerate to that of LDA and SGNS respectively). We found that the best results are obtained for $\lambda = 0.9$.

For node2vec baseline approach of word-node embedding, we varied the parameters p and q (BFS and DFS parameters) within a range of $[0.1, 5]$ and found that the best results on the MEN dataset are given for $p = 1$ and $q = 1$ (Grover and Leskovec, 2016). Another parameter in node2vec is the random walk length, l, for which the optimal value was found to be 80.

For word-node2vec, in addition to window size (k) and number of negative samples (NS), three more parameters are: i) α, i.e., the importance of the presence of a term relative to its informativeness (Equation 12, ii) β, the prior assigned to sampling from the 1-adjacent neighborhood, and iii) the size of the context sampled from the neighborhood, l (this is analogous to the random walk length parameter of node2vec). Instead of separately optimizing the parameters common to SGNS, we directly use the optimal values of $k = 10$ and $NS = 5$ for word-node2vec. The optimal results of the additional parameters, tuned on the MEN dataset, are shown in Table 3.

6 Results

Word Similarity Prediction. Table 3 shows the results obtained by the competing methods on the word similarity prediction task. It can be seen that Glove turns out to be relatively ineffective in modeling the semantic representations of words

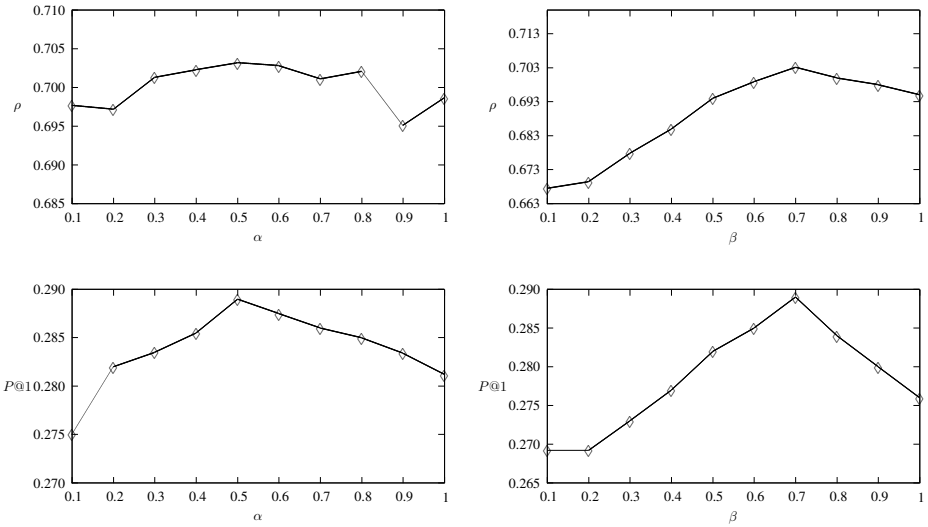

Figure 1: Parameter sensitivity of word-node2vec on word prediction (left column) and word analogy (right column) tasks using WSIM (top row) and MSR (bottom row) datasets.

Method	Accuracy (P@1)		
	Google	MSR	SemEval'12
SGNS	0.5615	0.2777	0.1460
Glove	0.4841	0.2485	0.1419
FastText	0.4930	0.2607	**0.1592**
ELMO	**0.5986**	0.2789	0.1439
LDA	0.0578	0.0158	0.0596
SGNS-LDA	0.5491	0.2776	0.1413
Node2vec	0.5588	0.2785	0.1427
Word-node2vec	0.5627	**0.2890**	0.1464

Table 4: Word analogy results.

Method	Accuracy		
	AP	BLESS	ESSLI_{2b}
SGNS	0.6194	0.7500	0.7500
Glove	0.6343	0.7200	0.7250
FastText	0.6119	**0.7950**	0.7250
ELMO	0.6368	0.7350	0.7500
LDA	0.3383	0.3900	0.6500
SGNS-LDA	0.5796	0.7850	**0.7750**
Node2vec	0.6355	0.7500	0.7350
Word-node2vec	**0.6393**	**0.7950**	**0.7750**

Table 5: Concept categorization results.

as compared to human judgments. SGNS performs significantly better and the settings trained on MEN dataset generalize well on the WSIM-353 dataset as well. LDA performs rather poorly indicating that only global co-occurrences can lead to noisy representations of words. FastText performs worse as compared to SGNS. It is worth mentioning that the performance of ELMO is disappointing on this task of semantic similarity prediction, because of the most likely reason that it better learns vector representations of word in the presence of a context.

A linear combination of SGNS and LDA (Equation 1 with $\lambda = 0.9$) does not perform better than SGNS, which means that a simple way of combining the embedded representations obtained individually with local and non-local approaches does not work well.

The node2vec approach of embedding nodes of the word-nodes graph constructed as per the description of Section 4 relies on a random walk based construction of the context of a word node. This random walk based context construction is only able to improve the SGNS results slightly, indicating that random walks can introduce noise in the contexts of word-nodes.

The word-node based graph construction (incorporating local and non-local co-occurrences in a principled way) works particularly well in conjunction with the stratified sampling based approach of selecting context words from the κ-neighborhood. The optimal value of $\alpha = 0.5$ suggests that document-level co-occurrences should be computed by assigning equal importance to term presence and informativeness. A value of $\beta = 0.7$ confirms the hypothesis that more emphasis should be put on direct co-occurrences.

Word Analogy and Concept Categorization. Similar trends are observed in the word analogy and concept categorization tasks in Tables 4 and 5 respectively. Relatively higher improve-

Rank	SGNS		word-node2vec	
1	albums	0.929	albums	0.926
2	selftitled	0.885	selftitled	0.883
3	rerecorded	0.868	rerecorded	0.863
4	promotional	0.815	**released**	0.852
5	reissue	0.790	**song**	0.810

Table 6: Nearest neighbors of the word 'album' obtained by SGNS and word-node2vec.

ments with word-node2vec are noted for the MSR analogy task (comprised of syntactic categories). Among the baseline approaches, both node2vec and SGNS-LDA work well on the concept categorization task. However, the performance improvements are inconsistent across datasets, e.g. SGNS-LDA performs well on ESSLI_{2b} and poorly on AP. Our proposed method configured on the MEN dataset works consistently well across all datasets, which indicates that word-node2vec can generalize well for different tasks.

As a side observation, we note that ELMO performs well for the analogy and concept categorization tasks (yielding the best results in particular on the Google analogy dataset). Although the results are not directly comparable because of differences in the dimensionality of the vectors and also in the collection of documents used in the pre-trained ELMO vectors (Billion word benchmark as against DBPedia in our case), it could possibly be reasoned that the additional contextual information of the ELMO vectors turns out to be useful for in the analogy task.

Embedding Examples. Table 6 shows an example of the change in the neighbourhood of a sample word in the embedded space obtained by SGNS and word-node2vec. It can be seen from the table that word-node2vec is able to push relevant words, such as 'released' and 'song' within the top 5-NN of the word 'album'. Although the words 'promotional' and 'reissue' are related to 'album', the semantic association of 'released' and 'song' with 'album' is apparently higher. We found that the word 'song' occurs in the local context of the word 'album' only $133,494$ number of times out of a total number of $177,487$ instances of the word 'album'. This means that a significant percentage of times (almost 25%), 'song' co-occurs with 'album' at a document-level. Our embedding algorithm is able to leverage this information by making the vector for 'song' closer to 'album'.

Sensitivity Analysis. Tables 3-5 show word-node2vec results with optimal parameter settings. We now investigate the effect of varying these parameters on each individual evaluation task. We observe that both term presence and term informativeness are important to model document-level co-occurrences as seen from the fact that the ρ and accuracy values decrease as α gets close to 0 or 1 (the 1st and 3rd plots from the left of Figure 1). Similarly, it can be seen that the results tend to improve with higher values of β, which confirms that direct associations between words in the word-node graph are more important than transitive ones (2nd plot from the left and the rightmost plot of Figure 1). However, second-order transitive associations are still important because the results tend to decrease for β close to 1.

7 Conclusions and Future work

We proposed a word embedding approach that leverages document-level non-local co-occurrences, in addition to the window-based local co-occurrences. We proposed a graph-based framework, in which words are represented as nodes and the edges between a pair of words reflect the degree of association between them. This association is a function of both the local and the document-level co-occurrences, which enables our approach to achieve 'the best of both worlds' in word embedding. Experiments show that our proposed method outperforms local approaches, namely word2vec, Glove and FastText, on a number of different tasks. Our approach also outperforms a naive black-box combination of embeddings obtained separately by local and document-level approaches. This proves the importance of addressing both these sources of information jointly in an embedding objective.

In future, we would like to explore ways of applying a similar graph based formalism for learning vectors for documents.

Acknowledgements

This work was supported by Science Foundation Ireland as part of the ADAPT Centre (Grant No. 13/RC/2106) (www.adaptcentre.ie). This work started as an internship during the first author's visit to IBM Research Lab, Ireland.

References

Abdulrahman Almuhareb and Massimo Poesio. 2005. Concept learning and categorization from the web. In *Proc. of COGSCI*, pages 103–108.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proc. of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49:1–47.

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):391–407.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proc. of NAACL HLT*, pages 1606–1615.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2014. Placing search in context: The concept revisited. In *Proc. of WWW 2014*, pages 406–414.

Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J. F. Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proc. of SIGIR'15*, pages 795–798.

Eric Gaussier and Cyril Goutte. 2005. Relation between plsa and nmf and implications. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 601–602.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proc. of the 22Nd ACM SIGKDD 2016*, pages 855–864.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proc. of CIKM '16*, pages 55–64.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296. Morgan Kaufmann Publishers Inc.

Stanislaw Jastrzebski, Damian Lesniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *CoRR*, abs/1702.02170.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

David A. Jurgens, Peter D. Turney, Saif M. Mohammad, and Keith J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proc. of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proc. of the Main Conference and the Shared Task, and Volume 2: Proc. of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 356–364.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185.

Stefan Evert. Marco Baroni and Alessandro Lenci. 2008. Esslli workshop on distributional lexical semantics. *ESSLLI Workshop on Distributional Lexical Semantics*, 101:1–70.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS 2013*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proc. of NAACL 2013*, pages 746–751.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proc. of AAAI'16*, pages 2786–2792.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP 2014*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL 2018*.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281. ACM.

Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth J. F. Jones. 2016. Word vector compositionality based relevance feedback using kernel density estimation. In *Proc. of CIKM'16*, pages 1281–1290. ACM.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with bayesian optimization for task-specific word representation learning. In *Proc. of ACL'16*, pages 130–139.