Physica A 523 (2019) 180–190

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Word sense induction using word embeddings and community detection in complex networks

Edilson A. Corrêa Jr. [a], Diego R. Amancio [a,b,*]

[a] Institute of Mathematics and Computer Science, University of São Paulo (USP) São Carlos, São Paulo, Brazil
[b] School of Informatics, Computing and Engineering, Indiana University Bloomington,, IN 47408, USA

HIGHLIGHTS

- A method to represent occurrences of words as networks is proposed.
- A method to cluster word senses is proposed.
- Community detection methods are used to cluster word senses.
- We studied the word sense induction task as a language network.

ARTICLE INFO

Article history:
Received 18 September 2018
Received in revised form 10 January 2019
Available online 22 February 2019

Keywords:
Word sense induction
Language networks
Complex networks
Word embeddings
Community detection
Word sense disambiguation
Semantic networks

ABSTRACT

Word Sense Induction (WSI) is the ability to automatically induce word senses from corpora. The WSI task was first proposed to overcome the limitations of manually annotated corpus that are required in word sense disambiguation systems. Even though several works have been proposed to induce word senses, existing systems are still very limited in the sense that they make use of structured, domain-specific knowledge sources. In this paper, we devise a method that leverages recent findings in word embeddings research to generate *context embeddings*, which are embeddings containing information about the semantical context of a word. In order to induce senses, we modeled the set of ambiguous words as a complex network. In the generated network, two instances (nodes) are connected if the respective *context embeddings* are similar. Upon using well-established community detection methods to cluster the obtained *context embeddings*, we found that the proposed method yields excellent performance for the WSI task. Our method outperformed competing algorithms and baselines, in a completely unsupervised manner and without the need of any additional structured knowledge source.

1. Introduction

In recent years, language has been studied via many different approaches and techniques. An interesting feature of language is its ability to convey multiple meanings. While such a characteristic is oftentimes useful to enrich a discourse, an ambiguous word may cause a deleterious effect in the automatic processing and classification of texts. The identification of the sense of a word corresponds to the identification of its meaning in a given context. For instance, the word "bear" might be related to a wild mammal in a given context. In a different context, it may mean "to endure" a difficult situation.

* Corresponding author at: Institute of Mathematics and Computer Science, University of São Paulo (USP) São Carlos, São Paulo, Brazil.
E-mail addresses: diego@icmc.usp.br, diego.raphael@gmail.com (D.R. Amancio).

https://doi.org/10.1016/j.physa.2019.02.032
0378-4371/

In this paper, we address the problem of identifying the meaning (senses) of words in the word sense disambiguation task [1].

The Word Sense Induction (WSI) task aims at inducing word senses directly from corpora [1] (i.e. sets of textual documents). Since it has been shown that the use of word senses (rather than word forms) can be used to improve the performance of many natural language processing applications, this task has been continuously explored in the literature [1–3]. In a typical WSI scenario, automatic WSI systems identify the activated sense of a word in a given context, using a variety of features [1]. This task is akin to the word sense disambiguation (WSD) problem [4], as both induction and disambiguation requires the effective identification of the sense being conveyed. While WSD systems require, in some cases, large corpora of annotated senses, the inductive counterpart (also referred to as unsupervised WSD) does not rely upon any manual annotation [5], avoiding thus the knowledge acquisition bottleneck problem [6].

Analogously to what occurs in supervised disambiguation, WSI techniques based on machine learning represent the state-of-the art, outperforming linguistic-based/inspired methods. Several machine learning methods address the sense identification problem by characterizing the occurrence of an ambiguous word and then grouping together elements that are similar [1,2]. The characterization is usually done with the syntactic and semantic properties of the word, and other properties of the context where it occurs. Once a set of attributes for each occurrence of the ambiguous word is defined, a clustering/grouping method can be easily applied [1,2].

Textual contexts are usually represented by vector space models [7]. In such models, the context can be represented by the frequency of the words occurring in a given text interval (defined by a window length). Such a representation and its variants are used in several natural language processing (NLP) applications, owing to its simplicity and ability to be used in conjunction with machine learning methods. The integration of machine learning methods and vector space models is facilitated mostly because machine learning methods typically receive structured data as input. Despite of the inherent simplicity of bag-of-word models, in recent years, it has been shown that they yield a naive data representation, a characteristic that might hamper the performance of classification systems [8]. In order to overcome these problems, a novel vector representation – the *word embeddings* model – has been used to represent texts [9]. The *word embeddings* representation, also referred to as *neural word embeddings*, are vectors learnt from neural networks in particular language tasks, such as language modeling. The use of vector representations has led to an improvement in performance of several NLP applications, including machine translation, sentiment analysis and summarization [8,10–12]. In the current paper, we leverage the robust representation provided by word embeddings to represent contexts of ambiguous words.

Even though distributional semantic models have already been used to infer senses [13], other potential relevant features for the WSI problem have not been combined with the rich contextual representation provided by the *word embeddings*. For example, it has been shown that the structural organization of the context in bag-of-words models also provides useful information for this problem and related textual problems [14,15]. For this reason, in this paper, we provide a framework to combine the word embeddings representation with a model that is able to grasp the structural relationship among contexts. More specifically, here we address the WSI problem by explicitly representing texts as a complex network [16], where words are linked if they are *contextually* similar (according to the word embeddings representation). By doing so, we found out that the contextual representation is enhanced when the relationship among context words is used to cluster contexts in traditional community detection methods [17,18]. The advantage of using such methods relies on their robustness and efficiency in finding natural groups in highly clustered data [17]. Despite of making use of limited deep linguistic information, our method outperformed several baselines and methods that participated in the SemEval-2013 Task 13 [1].

The paper is organized as follows. Section 2 presents some basic concepts and related work. Section 3 presents the details of the proposed WSI method. Section 4 presents the details of the experiments and results. Finally, in Section 6 we discuss some perspectives for further works.

2. Background and related work

The WSI task was originally proposed as an alternative to overcome limitations imposed by systems that rely on sense inventories, which are manually created. The essential idea behind the WSI task is to group instances of words conveying the same meanings [4]. In some studies, WSI methods are presented as unsupervised versions of the WSD task, particularly as an effort to overcome the knowledge acquisition bottleneck problem [6]. Although some WSI methods have emerged along with the first studies on WSD, a comprehensive evaluation of methods was only possible with the emergence of shared tasks created specifically for the WSI task [1,2,19,20].

Several WSI methods use one of the three following methodologies: (i) word clustering; co-occurrence graphs; and (iii) context clustering [4]. Word clustering methods try to take advantage of the semantical similarity between words, a feature that is usually measured in terms of syntactical dependencies [21,22]. The approach based on co-occurrence graphs constructs networks where nodes represent words and edges are the syntactical relationship between words in the same context (sentence, paragraph or larger pieces of texts). Given the graph representation, word senses are identified via clustering algorithms that use graphs as a source of information [23,24]. The framework proposed in this manuscript uses the graph representation, however, links are established using a robust similarity measure based on *word embeddings* [25]. Finally, context clustering methods model each occurrence of an ambiguous word as a context vector, which can be clustered by traditional clustering methods such as Expectation Maximization and k-means [26]. Differently

from graph approaches, the relationship between context words is not explicitly considered in the model. In [12], the authors explore the idea of context clustering, but instead of using context vectors based on the traditional vector space model (bag-of-words), they propose a method that generates embeddings for both ambiguous and context words. The method – referred to as Instance-Context Embeddings (ICE) – leverages neural word embeddings and correlation statistics to compute high quality word context embeddings [12]. After the embeddings are computed, they are used as input to the *k*-means algorithm in order to obtain clusters of similar senses. A competitive performance was reported when the method was evaluated in the SemEval-2013 Task 13 [20]. Despite its ability to cluster words conveying the same sense, the performance of the ICE system might be very sensitive to the parameter k in the *k*-means method (equivalently, the number of senses a word can convey), which makes it less reliable in many applications where the parameter is not known a priori.

In the present work, we leverage word embeddings to construct complex networks [14,27–30]. Instead of creating a specific model that generates context embeddings, we use pre-trained embeddings and combine them to generate new embeddings. The use of pre-trained word embeddings is advantageous because these structures store, in a low-cost manner, the semantical contextual information of words trained usually over millions of texts. Another distinguishing characteristic of our method is that it explores three successful strategies commonly used in WSI. Firstly, we use semantic information by modeling words via word embeddings. We then make use of complex networks to model the problem. Finally, we use community detection algorithms to cluster instances conveying the same sense. The proposed strategy is also advantageous because the number of senses do not need to be known a priori, since the network modularity can be used to suggest the number of clusters providing the best partition quality [18]. The superiority of clustering in networked data over traditional clustering methods has also been reported in the scenario of semantical classification of words.

3. Overview of the technique

The proposed method can be divided into three stages: (i) context modeling and context embeddings generation, (ii) network modeling and (iii) sense induction. These steps are described respectively in Sections 3.1–3.3.

3.1. Context modeling and context embeddings generation

Several ways of representing the context have been widely stressed by the literature [4]. Some of them consist of using vector space models, also known as bag-of-words, where features are the words occurring in the context. Other alternative is the use of linguistic features, such as part-of-speech tagging and collocations [31]. Some methods even propose to combine two or more of the aforementioned representations [32].

In recent years, a set of features to represent words – the word embeddings model – has become popular. Although the *representation of words as vectors* has been widely adopted for many years [4], only recently, with the use of neural networks, this type of representation really thrived. For this reason, from now on word embeddings refer only to the recent word representations, such as *word2Vec* and *GloVe* [33,34]. As in other areas of NLP, word embeddings representations have been used in disambiguation methods, yielding competitive results [35].

In this work, we decided to model context using word embeddings, mostly because acquiring and creating this representation is a reasonable easy task, since they are obtained in a unsupervised way. In addition, the word embeddings model has been widely reported as the state-of-the art word representation [36]. First introduced in [37], the neural word embeddings is a distributional model in which words are represented as continuous vectors in an ideally semantic space. In order to learn these representations, [37] proposed a feed-forward neural network for language modeling that simultaneously learns a distributed representation for words and the probability function for word sequences (i.e., the ability to predict the next word given a preceding sequence of words). Subsequently, in [38], the authors adapted this concept into a deep neural architecture, which has been applied to several NLP tasks, such as part-of-speech tagging, chunking, named entity recognition, and semantic role labeling [38,39].

A drawback associated to the architectures devised in [37,38] is their high computational cost, which makes them prohibitive in certain scenarios. To overcome such a complexity, in [33,40], the authors proposed the *word2vec* representation. The *word2vec* architecture is similar to the one created in [37]. However, efficient algorithms were proposed so as to allow a fast training of word embeddings. Rather than being trained in the task of language modeling, two novel tasks were created to evaluate the model: the prediction of a word given its surrounding words (continuous bag-of-words) and the prediction of the context given a word (skipgram).

The word embeddings (i.e. the *vector representation*) produced by *word2vec* have the ability to store syntactic and semantic properties [40]. In addition, they have geometric properties that can be explored in different ways. An example is the *compositionality* property, stating that larger blocks of information (such as sentences and paragraphs) can be represented by the simple combination of the embeddings of their words [33,40]. In this work, we leverage this property to create what we define as *context embeddings*. More specifically, we represent an ambiguous word by combining the embeddings of all words in its context (neighboring words in a window of size w) using simple operations such as addition.

Fig. 1 shows a representation of the process of generating the embeddings of a given occurrence of an ambiguous word. In the first step, we obtain each of the word vectors representing the surrounding words. Particularly, in the current study, the embeddings were obtained from the study conducted in [33,40]. The method used to obtain the embeddings

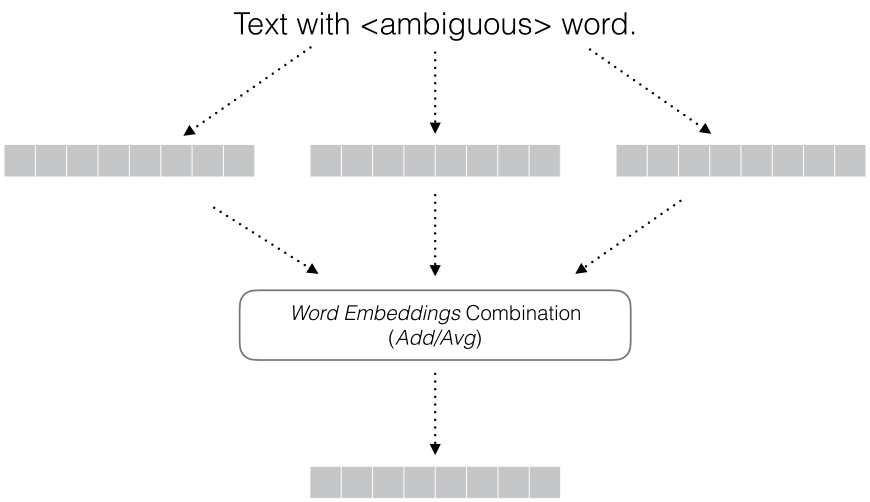

Fig. 1. Example illustrating how the context can be characterized from individual word embeddings. Given the word vectors representing the word appearing in the context, we combine those vectors to obtain a single embedding representing the context around the ambiguous word.

is the *word2vec* method, in the skipgram variation [33]. The training phase was performed using *Google News*, a corpus comprising about 100 billion words. As proposed in [33,40], the parameters for obtaining the methods were optimized considering semantical similarity tasks. After obtained individual embeddings representing each word in the considered context, such structures are combined into a single vector, which is intended to represent and capture the semantic features of the context around the target word. Here we adopted two distinct types of combination: by (i) addition; and (ii) averaging.

Let w_i be an ambiguous word (i.e. an ordered set of symbols from some alphabet), where i represents that the word is at the ith position in the considered text. Given the occurrence of w_i in a context ($\mathbf{c}_i$) comprising ω words surrounding w_i, i.e. $\mathbf{c}_i = [w_{i-\omega/2}, \ldots, w_{i-1}, w_i, w_{i+1} \ldots w_{i+\omega/2}]^\intercal$, the context embedding ($\mathbf{c}_i$) of w_i obtained from addition is

$$\mathbf{c}_i = \sum_{\substack{j=-\omega/2 \\ j\neq 0}}^{+\omega/2} \mathbf{w}_{i+j}, \tag{1}$$

where $\mathbf{w}_j$ is the embedding (i.e. the vector representation) of the jth word in $\mathbf{c}_i$. In other words, the context of a word is given by the composition of the semantic features (word embeddings) associated to the neighboring words. This approach is hereafter referred to as CNN-ADD method.

In the average strategy, a normalizing term is used. Each dimension of the embedding is divided by the number of words in the context set. Let $l = |\mathbf{c}_i|$ be size of the context. The average context embedding is defined as:

$$\mathbf{c}_i = \sum_{\substack{j=-\omega/2 \\ j\neq 0}}^{+\omega/2} \frac{\mathbf{w}_{i+j}}{l}. \tag{2}$$

This approach is hereafter referred to as CNN-AVG method.

While differences between CNN-ADD and CN-AVG are not evident when computing distances with the cosine similarity, differences arise when the Euclidean distance is used to construct the network. This happens because not all similarity (or distance) measurements are scale invariant. Nonetheless, the results for the task considering variations with and without the scale factor are similar, as shown in the results.

3.2. Modeling context embeddings as complex networks

Modeling real-valued vectors into complex networks is a task that can be accomplished in many ways. Here we represent the similarity between contexts as complex networks, in a similar fashion as it has been done in previous works modeling language networks [16]. While in most works two words are connect if they are similar according to specific criteria, in the proposed model two context vectors are linked if the respective context embeddings are similar. Usually, two strategies have been used to connect nodes. In the k-NN approach, each node is connected to the k nearest (i.e. most similar) nodes. Differently, in the d-proximity method, a distance d is fixed and each node is connected to all other nodes with a distance equal or less than d [16].

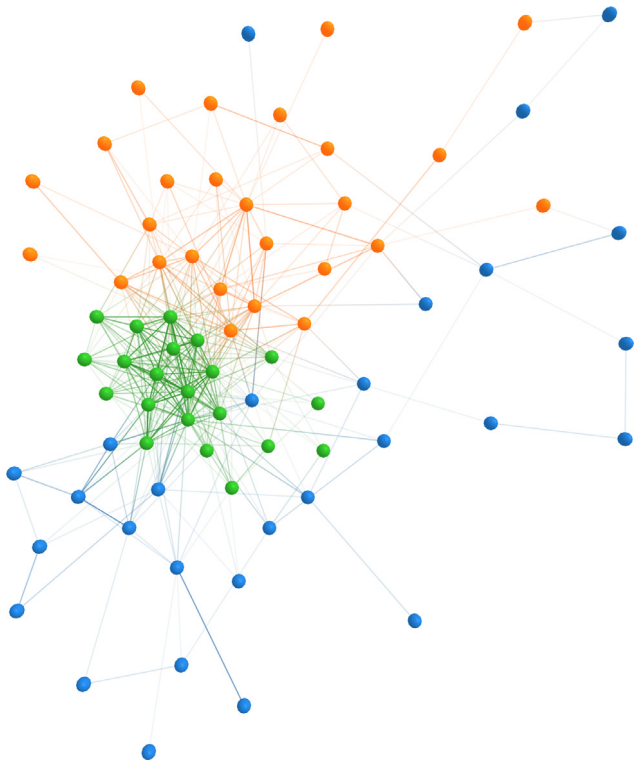

Fig. 2. Example of network obtained from the proposed model using $\omega = 10$ for the CN-ADD model. Each distinct color represents a different sense induced for the word "add". The visualization was obtained with the *networks3d* software [41].

In this work, similar to the approach adopted in [16], we generate complex networks from context embeddings using a k-NN approach. We have chosen this strategy because the network becomes connected with low values of k, thus decreasing the complexity of the generated networks. In addition, it has been shown that the k-NN strategy is able to optimize the modularity of the generated networks [16], an important aspect to our method. Both Euclidean and cosine were used as distance measurements. In the Euclidean case, the inverse of the distances was used as edges weight. In Fig. 2, we show the topology of a small network obtained from the proposed methodology for the word "add". Each node represents an occurrence of "add", which may convey three different senses in the considered dataset. Once the context vectors for each occurrence is obtained, they are linked by edges. To construct this visualization, we used $\omega = 10$ in the CNN-ADD model. Finally, senses are clustered via network community detection. Note that there is an evident separation among the three distinct senses.

3.3. Sense induction

Once the context embedding network is obtained, the Louvain community detection method [42] is applied to identify communities. Given the communities produced by the method, we define each community as a induced word sense. We have chosen the Louvain method because it is known to maintain reasonable computational costs [41] while maximizing the modularity [18]. We also have decided to use this method because it does not need any additional parameter definition to optimize the modularity function. The results obtained for other community detection method are provided in the Supplementary file. We decided not to show the results for these methods here because they are not significantly better than the ones obtained with the Louvain method.

To illustrate the process of identifying (clustering) the sense of ambiguous words, we show in Fig. 3 an example of the ambiguous word "bear", which may convey two senses in the example: (i) a verb with the meaning of enduring something; and (ii) a noun representing the wild mammal. The first step is to consider the embeddings of the context words. In the first sentence, the context word considered is "pain". In the second sentence, the considered context words are "out" and "woods". The representation of the context is then obtained by averaging the embeddings of the context words. Note that the embedding representing the ambiguous words in the second sentence is the average of the embeddings representing "out" and "woods". Once each occurrence of the ambiguous word is represented via embeddings, a network of similar embeddings is constructed and network community detection is used to discriminate senses.

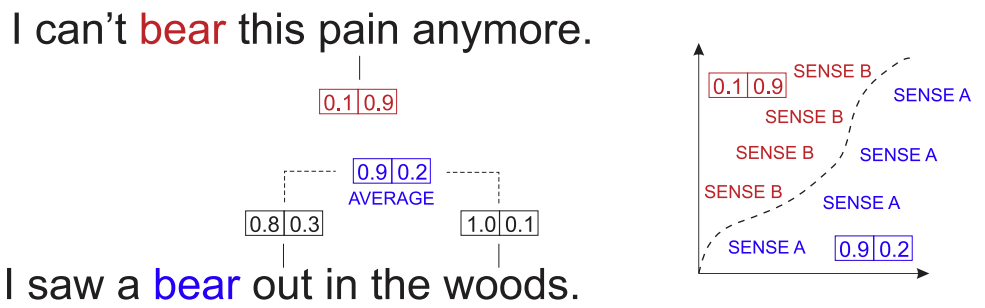

Fig. 3. Example of how senses are classified according to our methodology. In this example, the word "bear" can convey two different meanings. Note that the classification of senses relies on the embeddings of context words.

4. Corpora description

In this section, we present the Semeval-2013 corpora used to evaluate our method. The pre-trained word embeddings used here is also presented.

4.1. Semeval-2013 task 13 corpus

The SemEval-2013 data comprises 50 words. The number of instances of each word ranges between 22 and 100 instances. The dataset encompasses 4664 instances that were drawn from the *Open American National Corpus*. Each instance is a short piece of text surrounding an ambiguous word that came from a variety of literary genres. The instances were manually inspected to ensure that ambiguous words have at least one interpretation matching one of the WordNet senses.

Following the SemEval-2013 Task 13 proposal [20], we applied a two-part evaluation setting. In the first evaluation, the induced senses are converted to WordNet 3.1 senses via a mapping procedure and then these senses are used to perform WSD. The output of WSD is evaluated according to the following three aspects:

1. *Applicability*: this aspect is used to compare the set of senses provided by the system and the gold standard. The applicability criteria, in this context, is measured with the traditional Jaccard Index, which reaches its maximum value when the set of obtained senses and the gold standard are identical.
2. *Senses ranking*: the set of applicable senses for an ambiguous word might consider a different degree of applicability for distinct senses. For this reason, in addition to only considering which senses are applicable, it is also important to probe if the rank of importance assigned for the senses follows the rank defined by the gold standard. The agreement in applicability importance is measured using the positionally-weighted Kendall's τ (K_{δ}^{sim}) [20].
3. *Human agreement*: this measurement considers the WSI task as if it were tackled in the information retrieval scenario. In other words, the context of an ambiguous word is a query looking for all senses of the word. The expected retrieved information is the set of all applicable senses, which should be scored and ranked according to the applicability values of the word senses. This criterium was measured using the traditional Normalized Discounted Cumulative Gain (WNDCG) metric, as suggested by the literature [20].

All above measurements generate values between 0 and 1, where 1 means total agreement with the gold standard. As suggested in similar works, the final score is defined using the F1 measure between each of the objective's measure and the recall [20]. In this case, the recall measures the average score for each measure across all instances, even the ones that were not labeled by the WSD system.

In the second evaluation, the induced senses are compared with a sense inventory through clustering comparisons. In this case, the WSI task is considered as a clustering task and, because each word may be labeled with multiple senses, fuzzy measures are considered. In [20], the authors propose the use of the following fuzzy measures:

1. *Fuzzy B-Cubed*: this measurement summarizes the performance per instance providing an estimate of how well the WSI system would perform on a new corpus with a similar sense distribution.
2. *Fuzzy Normalized Mutual Information*: this index measures the quality of the produced clusters based on the gold standard. Differently from the Fuzzy B-Cubed score, the Fuzzy Normalized Mutual Information is measured at the cluster level, giving an estimate of how well the WSI system would perform independently of the sense distribution of the corpus.

4.2. Word embeddings

The pre-trained word embeddings[1] used in this study was trained as a part of the *Google News* dataset, which is composed of approximately 100 billion words. The model consists of three million distinct words and phrases, where each embedding is made up of 300 dimensions. All embeddings were trained using the *word2vec* method [33,40].

5. Results and discussion

Here we analyze the performance of the proposed methods (Section 5.1). In Section 5.2, we study the influence of the parameters on the performance of the methods based on complex network created from word embeddings.

5.1. Performance analysis

The results obtained by our model were compared with four baselines: (1) One sense, where all instances are labeled with the same sense; (2) 1c1inst, where each instance is defined as a unique sense; (3) SemCor MFS, where each instance is labeled with the most frequent sense of the lemma in the SemCor corpus; and (4) SemCor Ranked Senses, where each instance is labeled with all possible senses for the instance lemma, and each sense is ranked based on its frequency in the SemCor corpus. We also compared our method with the algorithms that participated in the SemEval-2013 shared task. More specifically, in this task, nine systems were submitted by four different teams. The AI-KU team submitted three WSI systems based on lexical substitution [43]. The University of Melbourne (Unimelb) team submitted two systems based on a Hierarchical Dirichlet Process [44]. The University of Sussex (UoS) team submitted two systems relying on dependency-parsed features [45]. Finally, the La Sapienza team submitted two systems based on the Personalized Page Rank applied to the WordNet in order to measure the similarity between contexts [46].

In the proposed method, considering the approaches to generate context embeddings, the general parameter to be defined is the context window size ω. We used the values $\omega = \{1, 2, 3, 4, 5, 7, 10\}$ and the full sentence length. In the network modeling phase, context embeddings are transformed into networks. No parameters are required for defining the *fully-connected* model that generates a fully connected embeddings network. In the k-NN model, however, the k value must be specified. We used $k = \{1, 5, 15\}$.

Testing all possible combinations of parameters in our method resulted in 95 different systems. For simplicity's sake, only the systems with best performance in the evaluation metrics are discussed in this section. Additional performance results are provided in the Supplementary Information. In the following tables the proposed models will be presented by acronyms that refer to the context features used: CN-ADD (Addition) or CN-AVG (Average). CN-ADD/AVG denotes that both systems displayed the same performance. When the ω column is empty, the full context (i.e. the full sentence) was used. Otherwise, the value refers to the context window. The k column refers to the value of the parameter k in the k-NN approach used to create the networks. When k is empty, the *fully-connected* model was used; otherwise, the value refers to the connectivity of the k-NN network.

Three major evaluations were carried out. In the first evaluation, methods were compared using all instances available in the shared task. The obtained results for this case are shown in Table 1. Considering the detection of which senses are applicable (see Jacc. Ind. column), our best methods outperformed all participants of the shared task, being only outperformed by the SemCor MFS method, a baseline known for its competitiveness [47]. Considering the criterium based on senses rank (as measured by the positionally-weighted Kendall's τ (K_δ^{sim})), our best methods also outperformed all competing systems, including the baselines. In the quantification of senses applicability (WNDCG index), our best methods are close to the participants; however, it is far from the best baseline (SemCor Ranked). Considering the cluster evaluation metrics, our method did not overcome the best baselines, but the same occurred to all participants of the SemEval task. Still, the proposed method outperformed various other methods in the clusters quality, when considering both Fuzzy NMI and Fuzzy B-Cubed criteria. It is interesting to note that, in this case, the best results were obtained when the fully (weighted) connected network was used to create the networks. In other words, the consideration of all links, though more computationally expensive, seems to allow a better discrimination of senses in this scenario.

In the second evaluation, only instances labeled with just one sense were considered. The obtained results are shown in Table 2. Considering F1 to evaluate the sense induction performance, our method outperformed all baselines, but it could not outperform the best participants methods. In the cluster evaluation, conversely, our best method displayed the best performance when compared to almost all other participants. Only two methods (One Sense and SemCor MFS) outperformed our CN approach when considering the instance performance evaluation (as measured by the Fuzzy B-Cubed index). Regarding the best k used to generate networks, we have found that, as in the previous case, in most of the configuration of parameters, the best results were obtained when the fully connected network was used.

In the last assessment, only instances labeled with multiple senses were considered in the analysis. The obtained results are shown in Table 3. Considering the criterium based on ranking senses and quantifying their applicability, our method have had only results close to the participants and below the best baselines. However, our methods outperformed all participants in the detection of which senses are applicable (see Jaccard Index) and in both cluster evaluation criteria. Once again, most of the best results were obtained for a fully connected network in the k-NN connectivity method.

[1] code.google.com/archive/p/word2vec/.

Table 1
Performance of our best methods evaluated using all instances available in the shared task. The best results are highlighted in bold. Note that, for several criteria, the CN-based method outperformed other traditional approaches.

			WSD F1			Cluster comparison	
System	ω	k	Jaccard	K_δ^{sim}	WNDCG	Fuzzy NMI	Fuzzy B-Cubed
CN-ADD/AVG	10	–	**0.273**	**0.659**	0.314	0.052	0.452
CN-ADD/AVG	5	–	0.266	0.650	**0.316**	0.056	0.457
CN-ADD	2	–	0.252	0.588	0.293	**0.061**	0.373
CN-ADD/AVG	4	1	0.235	0.634	0.294	0.039	**0.485**
One sense	–	–	0.192	0.609	0.288	0.0	0.623
1c1inst	–	–	0.0	0.0	0.0	0.071	0.0
SemCor MFS	–	–	0.455	0.465	0.339	–	–
SemCor Ranked	–	–	0.149	0.559	0.489	–	–

Table 2
Performance of our best methods evaluated using instances that were labeled with just one sense. Best results are marked in bold. Note that the proposed CN approach outperforms traditional approaches when using both F1 and Fuzzy NMI criteria. The results for the SemCor Ranked are not shown because, in the analysis considered only one possible sense, SemCor Ranked and SemCor MFS are equivalent.

System	ω	k	F1	Fuzzy NMI	Fuzzy B-Cubed
CN-ADD	4	–	**0.592**	0.048	0.426
CN-ADD	2	–	0.554	**0.049**	0.356
CN-ADD/AVG	4	1	0.569	0.031	**0.453**
One sense	–	–	0.569	0.0	0.570
1c1inst	–	–	0.0	0.018	0.0
SemCor MFS	–	–	0.477	0.0	0.570

Table 3
Performance of our best methods evaluated using instances that were labeled with multiple senses. Best results are marked in bold.

			WSD F1			Cluster comparison	
System	ω	k	Jaccard	K_δ^{sim}	WNDCG	Fuzzy NMI	Fuzzy B-Cubed
CN-ADD/AVG	4	5	**0.473**	0.564	0.258	0.018	0.126
CN-ADD/AVG	7	1	0.438	**0.604**	0.257	**0.040**	0.131
CN-ADD/AVG	10	–	0.464	0.562	**0.263**	0.021	**0.137**
CN-ADD/AVG	4	1	0.441	0.595	0.256	**0.040**	0.129
One sense	–	–	0.387	0.635	0.254	0.0	0.130
1c1inst	–	–	0.0	0.0	0.0	0.300	0.0
SemCor MFS	–	–	0.283	0.373	0.197	–	–
SemCor Ranked	–	–	0.263	0.593	0.395	–	–

Overall, the proposed CN-based approach displayed competitive results in the considered scenarios, either compared to baselines or compared to the participating systems. The use of addition and averaging to generate context embeddings turned out to be equivalent in many of the best obtained results, when considering the same parameters. It is also evident from the results that the performance of the proposed method varies with the type of ambiguity being tackled (single sense vs. multiple sense). Concerning the variation in creating the embedding networks, it is worth mentioning that the *fully-connected* model displayed the best performance in most of the cases. However, in some cases the k-NN model also displayed good results for particular values of k. Concerning the definition of the context window size, no clear pattern could be observed in Tables 1–3. This means that the context size might depend on either the corpus some property related to the specificities of the ambiguous word. A further analysis of how the method depends on the parameters is provided in the next section.

5.2. Parameter dependence

In this section, we investigate the dependency of the results obtained by our method with the choice of parameters used to create the network. In Fig. 4, we show the results obtained considered three criteria: F1, NMI and Fuzzy B-Cubed. Subfigures (a)–(c) analyze the performance obtained for different values of k, while subfigure (d)–(f) show the performance obtained when varying the context size ω. The dashed lines represent the performance obtained when the *fully-connected* strategy is used ((a)–(c)) or the full context of the sentence is used ((d)–(f)). No dashed lines are shown in (d) and (e) because the performance obtained with the full context is much lower than the performance values shown for different values of ω.

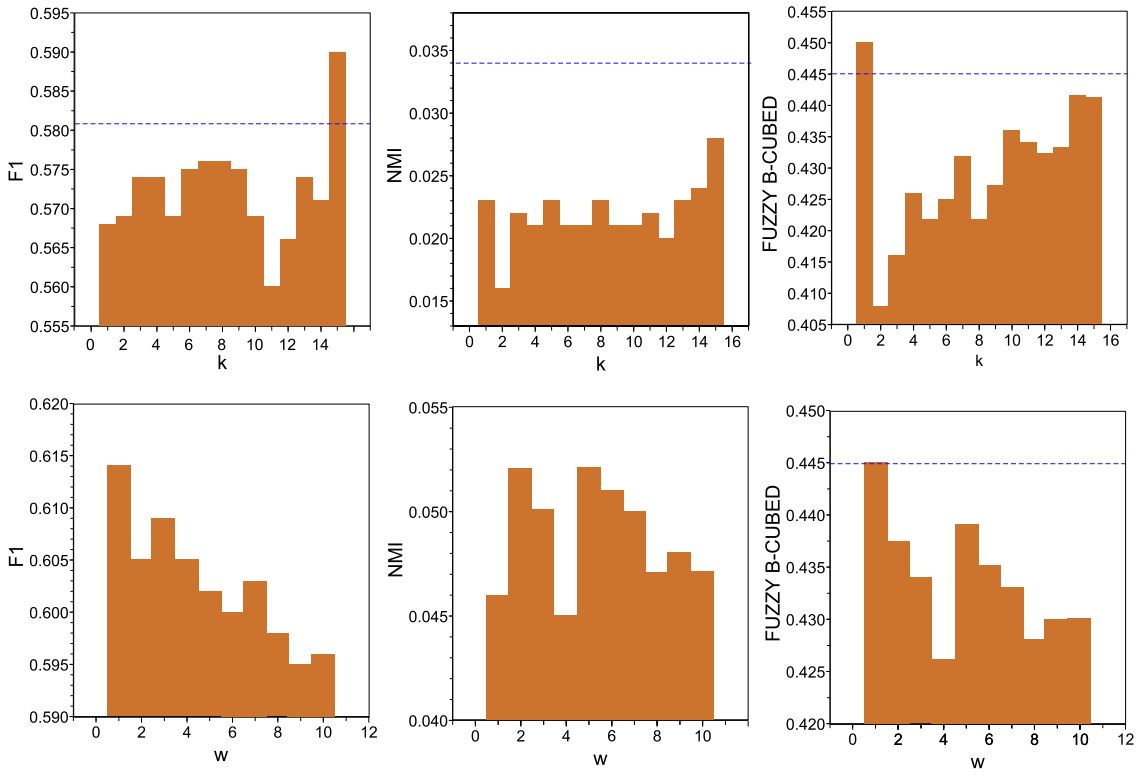

Fig. 4. Dependence of the performance results using different configuration of parameters. In all figures, we show the scenario allowing only one sense for the ambiguous word. In (a), (b) and (c); we analyze the behavior of the performance as a function of k when the full context of the sentence is considered. In (d), (e) and (f); we analyze the behavior of the performance for distinct values of ω when the fully-connected network is considered. The dashed lines represent the performance obtained with the full context (a–c) and the fully-connected network (f).

The variability of the performance with k reveals that, in general, a good performance can be obtained with high values of k. In (a), (b) and (c), excellent performances were obtained for $k = 15$. The fully-connected model also displayed an excellent performance in all three cases, being the best choice for the NMI index. These results confirm that the informativeness of the proposed model relies on both weak and strong ties, since optimized results are obtained mostly when all weighted links are considered. We should note, however, that in particular cases the best performance is achieved with a single neighbor connection (see Fig. 4(c)). Similar results can be observed for the other performance indexes, as shown in the Supplementary Information.

While the performance tends to be increased with high values of k, the best performance when ω varies seems to arise for the lowest values of context window. In (d) and (f), the optimum performance is obtained for $\omega = 1$. In (e), the NMI is optimized when $\omega = 2$. The full context only displays a good performance for the Fuzzy B-cubed measurement. Similar results were observed for the other measurements (see the Supplementary Information). Overall, the results showed that a low value of context is enough to provide good performance for the proposed model, considering both WSD-F1 and cluster comparison scenarios.

6. Conclusion

In this paper, we explored the concept of *context embeddings* modeled as complex networks to induce word senses via community detection algorithms. We evaluated multiple settings of our model and compared with well-known baselines and other systems that participated of the SemEval-2013 Task 13. We have shown that the proposed model presents a significant performance in both single and multiple senses multiple scenarios, without the use of annotated corpora, in a completely unsupervised manner. Moreover, we have shown that a good performance can be obtained when considering only a small context window to generate the embeddings. In a similar fashion, we have also found that, in general, a fully-connected and weighted network provides a better representation for the task. The absence of any annotation allows the use of the proposed method in a range of graph-based applications in scenarios where unsupervised methods are required to process natural languages.

As future works, we intend to explore the use of community detection algorithms that provide soft communities instead of the hard communities provided by most of the current methods. We also intend to explore the use of neural

language models to generate context embeddings in order to improve the quality of the context representation. Finally, we intend to integrate our methods with other natural language processing tasks [48–54] that might benefit from representing words as *context embeddings*.

Acknowledgments

E.A.C.Jr. acknowledges financial support from Google USA (Research Awards in Latin America grant) and CAPES-Brazil. D.R.A. acknowledges financial support from Google USA (Research Awards in Latin America grant) and São Paulo Research Foundation (FAPESP) Brazil (grant no. 2014/20830-0, 2016/19069-9 and 2017/13464-6).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.physa.2019.02.032.

References

[1] R. Navigli, D. Vannella, Semeval-2013 task 11: Word sense induction and disambiguation within an end-user application, in: Second Joint Conference on Lexical and Computational Semantics, * SEM, vol. 2, 2013, pp. 193–201.
[2] S. Manandhar, I.P. Klapaftis, D. Dligach, S.S. Pradhan, Semeval-2010 task 14: Word sense induction & disambiguation, in: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2010, pp. 63–68.
[3] K. Goyal, E.H. Hovy, Unsupervised word sense induction using distributional statistics, in: COLING, 2014, pp. 1302–1310.
[4] R. Navigli, Word sense disambiguation: A survey, ACM Comput. Surv. 41 (2009) 10.
[5] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, F. Ciravegna, Semantic annotation for knowledge management: requirements and a survey of the state of the art, Web Semant. Sci. Serv. Agents World Wide Web 4 (2006) 14–28.
[6] W.A. Gale, K.W. Church, D. Yarowsky, A method for disambiguating word senses in a large corpus, Comput. Humanit. 26 (1992) 415–439.
[7] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.
[8] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in: ACL, vol. 1, 2014, pp. 238–247.
[9] A. Mnih, K. Kavukcuoglu, Learning word embeddings efficiently with noise-contrastive estimation, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 2265–2273.
[10] K. Taghipour, H.T. Ng, Semi-supervised word sense disambiguation using word embeddings in general and specific domains, in: The 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2015, pp. 314–323.
[11] I. Iacobacci, M.T. Pilehvar, R. Navigli, Embeddings for word sense disambiguation: An evaluation study, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 897–907.
[12] M. Kågebäck, F. Johansson, R. Johansson, D. Dubhashi, Neural context embeddings for automatic discovery of word senses, in: Proceedings of NAACL-HLT, 2015, pp. 25–32.
[13] I. Iacobacci, M.T. Pilehvar, R. Navigli, Sensembed: learning sense embeddings for word and relational similarity, in: Proceedings of ACL, 2015, pp. 95–105.
[14] D.R. Amancio, O.N. Oliveira Jr., L.F. Costa, Unveiling the relationship between complex networks metrics and word senses, Europhys. Lett. 98 (2012) 18002.
[15] E. Corrêa Jr., A.A. Lopes, D.R. Amancio, Word sense disambiguation: A complex network approach, Inform. Sci. 442–443 (2018) 103–113.
[16] B. Perozzi, R. Al-Rfou, V. Kulkarni, S. Skiena, Inducing language networks from continuous space word representations, in: Complex Networks V, Springer, 2014, pp. 261–273.
[17] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.
[18] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (2006) 8577–8582.
[19] E. Agirre, A. Soroa, Semeval-2007 task 02: Evaluating word sense induction and discrimination systems, in: Proceedings of the 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics, 2007, pp. 7–12.
[20] D. Jurgens, I. Klapaftis, Semeval-2013 task 13: Word sense induction for graded and non-graded senses, in: Second Joint Conference on Lexical and Computational Semantics, * SEM, vol. 2, 2013, pp. 290–299.
[21] K. Sagae, A.S. Gordon, Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures, in: Proceedings of the 11th International Conference on Parsing Technologies, IWPT 09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 192–201.
[22] D. Lin, Automatic retrieval and clustering of similar words, in: Proceedings of the 17th International Conference on Computational Linguistics - Volume 2, COLING '98, Association for Computational Linguistics, Stroudsburg, PA, USA, 1998, pp. 768–774.
[23] D. Widdows, B. Dorow, A graph model for unsupervised lexical acquisition, in: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2002, pp. 1–7.
[24] J. Véronis, Hyperlex: lexical cartography for information retrieval, Comput. Speech Lang. 18 (2004) 223–252.
[25] Y. Liu, Z. Liu, T.-S. Chua, M. Sun, Topical word embeddings, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, pp. 2418–2424.
[26] M.Z. Rodriguez, C.H. Comin, D. Casanova, O.M. Bruno, D.R. Amancio, F.A. Rodrigues, L.F. Costa, et al., Clustering algorithms: a comparative approach, PLoS ONE 14 (2019) e0210236, http://dx.doi.org/10.1371/journal.pone.0210236.
[27] O.N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, N. Pržulj, Revealing the hidden language of complex networks, Sci. Rep. 4 (2014) 4547.
[28] Z.-K. Gao, X.-W. Zhang, N.-D. Jin, N. Marwan, J. Kurths, Multivariate recurrence network analysis for characterizing horizontal oil-water two-phase flow, Phys. Rev. E 88 (2013) 032910.
[29] F. Breve, L. Zhao, Fuzzy community structure detection by particle competition and cooperation, Soft Comput. 17 (2013) 659–673.
[30] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, J. Liu, Particle competition and cooperation in networks for semi-supervised learning, IEEE Trans. Knowl. Data Eng. 24 (2012) 1686–1698.
[31] Y. Wilks, M. Stevenson, Sense tagging: Semantic tagging with a lexicon, in: Tagging Text with Lexical Semantics: Why, What, and How? 1997.

[32] H. Sugawara, H. Takamura, R. Sasano, M. Okumura, Context representation with word embeddings for WSD, in: K. Hasida, A. Purwarianti (Eds.), Computational Linguistics, Springer Singapore, Singapore, 2016, pp. 108–119.
[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.
[34] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[35] T. Schnabel, I. Labutov, D. Mimno, T. Joachims, Evaluation methods for unsupervised word embeddings, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 298–307.
[36] J. Zhang, S. Liu, M. Li, M. Zhou, C. Zong, Bilingually-constrained phrase embeddings for machine translation, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 111–121.
[37] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.
[38] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.
[39] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[40] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[41] F.N. Silva, D.R. Amancio, M. Bardosova, L.d.F. Costa, O.N. Oliveira, Using network science and text analytics to produce surveys in a scientific topic, J. Infometrics 10 (2016) 487–502.
[42] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (2008) P10008.
[43] O. Baskaya, E. Sert, V. Cirik, D. Yuret, Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation, in: Second Joint Conference on Lexical and Computational Semantics, * SEM, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation, SemEval 2013, vol. 2, 2013, pp. 300–306.
[44] J.H. Lau, P. Cook, T. Baldwin, Unimelb: Topic modelling-based word sense induction, in: Second Joint Conference on Lexical and Computational Semantics, * SEM, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation, SemEval 2013, vol. 2, 2013, pp. 307–311.
[45] D. Hope, B. Keller, Uos: A graph-based system for graded word sense induction, in: Second Joint Conference on Lexical and Computational Semantics, * SEM, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation, SemEval 2013, vol. 2, 2013, pp. 689–694.
[46] E. Agirre, A. Soroa, Personalizing pagerank for word sense disambiguation, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2009, pp. 33–41.
[47] E. Agirre, O.L. De Lacalle, A. Soroa, Knowledge-based WSD on specific domains: Performing better than generic supervised WSD, in: Proceedings of the 21st International Jont Conference on Artifical Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009, pp. 1501–1506.
[48] D.R. Amancio, Comparing the topological properties of real and artificially generated scientific manuscripts, Scientometrics 105 (2015) 1763–1779.
[49] D.R. Amancio, Probing the topological properties of complex networks modeling short written texts, PLoS One 10 (2015) e0118394.
[50] K. Ban, M. Perc, Z. Levnajić, Robust clustering of languages across Wikipedia growth, Royal Soc. Open Sci. 4 (2017).
[51] H. Chen, X. Chen, H. Liu, How does language change as a lexical network? An investigation based on written Chinese word co-occurrence networks, PLoS One 13 (2018) e0187164.
[52] D. Yu, W. Wang, S. Zhang, W. Zhang, R. Liu, Hybrid self-optimized clustering model based on citation links and textual features to detect research topics, PLoS One 12 (2017) e0187164.
[53] D.R. Amancio, O.N. Oliveira Jr., L.d.F. Costa, On the use of topological features and hierarchical characterization for disambiguating names in collaborative networks, Europhys. Lett. 99 (2012) 48002.
[54] C. Akimushkin, D.R. Amancio, O.N. Oliveira Jr., Text authorship identified using the dynamics of word co-occurrence networks, PLoS One 12 (2017) e0170527.