Information Sciences 442–443 (2018) 103–113

Contents lists available at ScienceDirect

Information Sciences

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

Word sense disambiguation: A complex network approach

Edilson A. Corrêa Jr.[a], Alneu A. Lopes[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

ARTICLE INFO

Article history:
Received 8 September 2017
Revised 8 January 2018
Accepted 18 February 2018
Available online 19 February 2018

Keywords:
Complex networks
Network science
Text classification
Bipartite graphs
Bipartite networks
Word sense disambiguation

ABSTRACT

The word sense disambiguation (WSD) task aims at identifying the meaning of words in a given context for specific words conveying multiple meanings. This task plays a prominent role in a myriad of real world applications, such as machine translation, word processing and information retrieval. Recently, concepts and methods of complex networks have been employed to tackle this task by representing words as nodes, which are connected if they are semantically similar. Despite the increasingly number of studies carried out with such models, most of them use networks just to represent the data, while the pattern recognition performed on the attribute space is performed using traditional learning techniques. In other words, the structural relationships between words have not been explicitly used in the pattern recognition process. In addition, only a few investigations have probed the suitability of representations based on bipartite networks and graphs (bigraphs) for the problem, as many approaches consider all possible links between words. In this context, we assess the relevance of a bipartite network model representing both feature words (i.e. the words characterizing the context) and target (ambiguous) words to solve ambiguities in written texts. Here, we focus on semantical relationships between these two type of words, disregarding relationships between feature words. The adopted method not only serves to represent texts as graphs, but also constructs a structure on which the discrimination of senses is accomplished. Our results revealed that the adopted learning algorithm in such bipartite networks provides excellent results mostly when *local* features are employed to characterize the context. Surprisingly, our method even outperformed the support vector machine algorithm in particular cases, with the advantage of being robust even if a small training dataset is available. Taken together, the results obtained here show that the representation/classification used for the WSD problem might be useful to improve the semantical characterization of written texts without the use of deep linguistic information.

1. Introduction

The word sense disambiguation (WSD) task has been widely studied in the field of Natural Language Processing (NLP) [31]. This task is defined as the ability to computationally detect which sense is being conveyed in a particular context [37]. Although humans solve ambiguities in an effortlessly manner, this matter remains an open problem in computer science, owing to the complexity associated with the representation of human knowledge in computer-based systems [30]. The importance of the WSD task stems from its essential role in a variety of real world applications, such as machine translation

[*] Corresponding author.
E-mail addresses: diego@icmc.usp.br, diego.amancio@usp.br (D.R. Amancio).

https://doi.org/10.1016/j.ins.2018.02.047

[52], word processing [19], information retrieval and extraction [21,22,32,47,49,56]. In addition, the resolution of ambiguities plays a pivotal role in the development of the so-called semantic web [13].

Many approaches devised to solve ambiguities in texts employ machine learning methods, in which systems using supervised methods represent the state-of-the-art [37]. These methods usually rely on features extracted from the context of ambiguous (target) words, making contextual information a primordial element in the disambiguation process. However, the learning process in most of these methods only use representations that attempt to grasp the context and little or no explicit modeling of context is made. In this paper, we propose a new representation that explicitly models context and may be used as a underlying structure in the learning process. The representation used here consists of a bipartite network composed only of target and context words, while learning is carried out by a gradient descent method, which learns the relationship between the two types of words, allowing the induction of a model capable of performing supervised WSD.

Although networks/graphs have been employed in general pattern recognition methods [15,16,54] and, particularly in the analysis of the semantical properties of texts in several ways [3,7,11,28,33,35,43,50], the use of network models in the learning process has been restricted to a few works (see e.g. [44]). In fact, most of the current network models emphasize the relationship between *all* words of the document. As a consequence, a minor relevance has been given to the relationships between feature and target words. As we shall show, the adopted representation/learning method may improve the classification process when compared with well-known traditional/general purpose supervised algorithms hinging on traditional text representations. Remarkably, we have found that our method retains its discriminative power even when a considerable small amount of training instances is available. These results may indicate that the adopted method is an ideal candidate for state-of-the-art methods such as IMS [55], which at its core make use of traditional machine learning methods such as Support Vector Machines. We also applied our algorithm to two popular benchmarks in the area of WSD, namely *Senseval-3 English Lexical Sample Task* [34] and *SemEval-2007 Task 17 English Lexical Sample* [40]. Despite of making use of a simple superficial textual representation, in both datasets our method achieved intermediary positions.

The remainder of this paper is organized as follows. We first present a brief review of basic concepts employed in this paper and related works. We then present the details of the representation and algorithm used to tackle the word sense disambiguation task. The details of the experiments and the results concerning the accuracy and robustness of the method is also discussed. Finally, we present some perspectives for further works.

2. Related works

The word sense disambiguation task can be defined as follows. Given a document represented as a sequence of words $T = \{w_1, w_2, \ldots, w_n\}$, the objective is to assign appropriate sense(s) to all or some of the words $w_i \in T$. In other words, the objective is to find a mapping A from words to senses, such that $A(w_i) \subseteq \mathcal{S}_D(w_i)$, where $\mathcal{S}_D(w_i)$ is the set of senses encoded in a dictionary D for the word w_i, and $A(w_i)$ is the subset of appropriate senses of $w_i \in T$. One of the most popular approaches to tackle the WSD problem is the use of machine learning, since this task can be seen as a supervised classification problem, where senses represent the classes [37]. The attributes used in the learning methods are usually any informative evidence obtained from context and external knowledge sources. The latter approach is usually not common in practice because the creation of knowledge datasets demands a time-consuming effort, since the change in domains requires the recreation of new knowledge bases.

The generic WSD task can be distinguished into two types: *lexical sample* and *all-words* disambiguation. In the former, a WSD system is required to disambiguate a restricted set of target words. This is mostly done by supervised classifiers [37]. In the *all-words* scenario, the WSD system is expected to disambiguate all open-class words in a text. This task usually requires a wide-coverage of domains, and for this reason a knowledge-based system is usually employed. In this article, only the *lexical sample* task is considered.

The main step in any supervised WSD system is the representation of the context in which target words occur. The set of features employed typically are chosen to characterize the context in a myriad of forms [37]. The most common types of attributes used for this aim are:

- *local features*: the features of an ambiguous concept are a small number of words surrounding target words. The number of words representing the context is defined in terms of the window size ω. For example, if the context of the target word τ_ω is "$p_{-3}\ p_{-2}\ p_{-1}\ \tau_\omega\ p_{+1}\ p_{+2}\ p_{+3}$" and $\omega = 2$, then the words p_{-2}, p_{-1}, p_{+1} and p_{+2} are used as features.
- *topical features*: the features are defined as topics of a text or discourse, usually denoted in a bag-of-words representation;
- *syntatical features*: the features are syntactic cues and argument-head relations between the target word and other words within the same sentence; and
- *semantical features*: the features of a word are any semantic information available, such as previously established senses or domain indicators.

Using the aforementioned set of features, each word occurrence can be converted to a feature vector, which in turn is used as input in supervised classification algorithms. Typical classifiers employed for this task include decision trees [36], bayesian classifiers [20,36], neural networks [36] and support vector machines [20,26]. A well known state of the art system that uses a combination of the presented features is the "*It Makes Sense*" (IMS) method [55], which uses Support Vector

Machines as the standard classifier. This system also makes use of attributes derived from knowledge bases, allowing its application in both *all-words* and *lexical sample* tasks.

Another approach that has been used to address the WSD problem consists in the use of complex networks [6,38,45,53] and graphs [35]. For instance, the HyperLex algorithm [50] connects words co-occurring in paragraphs to establish similarity relations among words appearing in the same context. The frequency of co-occurrences is considered according to the following weighting scheme:

$$w_{ij} = 1 - \max\{P(w_i, w_j), P(w_j, w_i)\} \tag{1}$$

where $P(w_i, w_j) = f_{ij}/f_i$, f_i is the frequency of word i in the document and f_{ij} is the frequency of the co-occurrence of the words i and j. Then, this network is used to create a tree-like structure via recognition of central concepts, which represent all possible senses. To perform the classification, the distance of context words to the central concepts in the tree structure is computed to identify the most likely sense.

Using a different approach, [9] uses the local topological properties of co-occurrence networks to disambiguate target words. In this case, even though a significant performance has been found for particular target words, the optimal discrimination rate was obtained with traditional local features, suggesting thus that the overall discriminability could be improved upon combining features of distinct nature, as suggested by similar approaches [5,51].

Despite the numerous studies devoted to the WSD problem, this task remains an open problem in NLP, and currently it is considered one of the most complex problems in Artificial Intelligence [30]. Our contribution in this paper is the proposition of a new representation that explicitly models context that is used to perform sense discrimination. Unlike previous studies [9,50], the learning process takes place in the same structure used for representation, eliminating the need of hand-designed features. Despite its seemingly simplicity, we show that such representation captures, in a artlessly manner, informative properties of target words and their respective senses.

3. Overview of the technique

This section presents the approaches to represent the context of target words in a bipartite heterogeneous network. Here we also present the Inductive Model Based on Bipartite Heterogeneous Network (IMBHN) algorithm, which is responsible for inducing a classification model from the structure of a bipartite network [42,46].

3.1. Modelling word context as a bipartite heterogeneous network

Traditionally, the context of ambiguous words is represented in a vector space model, so that each target word is characterized by a vector. In this representation, each dimension of the vector corresponds to a specific feature. Alternatively, we may represent the data using a bipartite heterogeneous network. In this model, while the first layer comprises only feature words, the second only stores target words. In this paper, we focused on the analysis of *local* and *topical* attributes in the form of context, as such data are readily available on (or derivable from) any corpus. Note that, in this case, we have not used any knowledge dataset.

In the proposed strategy based on *topical* features, we create a set $\mathcal{T}$ of topical words. Then, each one becomes a distinct feature. As topical words, we considered the most frequent words of the dataset. The number of topical words, i.e. $|\mathcal{T}|$, is a free parameter. Given $\mathcal{T}$, the bipartite network is created by establishing a link between topical and target words whenever they co-occur in the same document.

In the proposed representation based on *local* features, each feature word surrounding the target word represents an attribute. For each instance of the target word in the text, we select the ω closest surroundings words to become a feature word (see definition in “Related works” section). The selected words are then connected to the target words by weighted edges.

3.2. Algorithm description

The IMBHN algorithm can be used in the context of any text classification task. If the objective is to classify distinct documents in a given number of classes, the bipartite network can be constructed so that nodes represent both terms and documents. In this general scenario, such representation is used to compute the relevance of specific terms for distinct document classes. In a similar fashion, in this study, we compute the relevance of *local*/*topical* features for each target word. Then, this relevance is used to infer word senses.

The algorithm employed for sense identification relies upon a network structure with two distinct layers: (i) a layer representing possible feature words (i.e. *local* or *topical* features), and (ii) a layer comprising all occurrences of the target word. The two layers are illustrated in Fig. 1. Edges are established across layers so that context words and distinct occurrences of the target word are connected. In addition, in the network representation, a weight relating each feature word to each target word is also established. The main components of the model are:

- w_{d_k,t_i}: the weight of the connection linking the kth target word and the ith feature word. In the strategy based on *topical* features, this weight is constant along the execution of the algorithm and, for a given instance T, is computed as

$$w_{d_k,t_i} = 1 - \delta(d_k, t_i)/l(T), \tag{2}$$

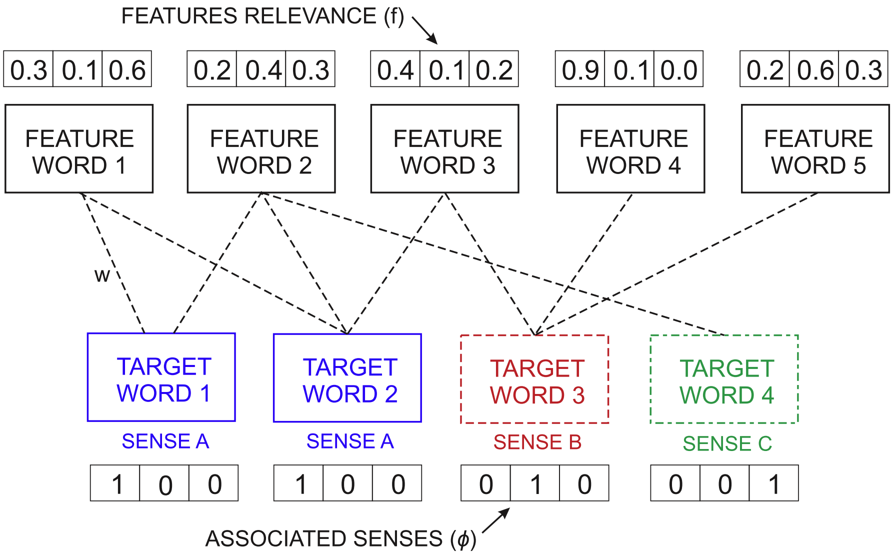

Fig. 1. Bipartite network structure used by the IMBHN algorithm. Note the existence of two layers: the layer comprising feature words and the layer comprising target words, which can be classified into three distinct senses (A, B and C). For each feature word, there exists a vector of features relevance whose element f_{t_i,c_j} denotes the relevance of ith feature word for the jth possible sense. The vectors below each target word represents the sense obtained in each iteration (i.e. ϕ_{d_k,c_j}).

where $\delta(d_k, t_i)$ denotes the distance between two words (i.e. the number of intermediary words) and $l(T)$ is the length of T (measured in terms of word counts). In the strategy based on *local* features, the weight of the links is given by the term frequency - inverse document frequency (tf-idf) strategy [31].

- f_{t_i,c_j}: let $\mathcal{C}$ be the set of possible classes (i.e. word senses). f_{t_i,c_j} represents the current relevance of the ith feature word ($t_i \in \mathcal{T}$) to the jth class ($c_j \in \mathcal{C}$). This value is initialized using a heuristic and then is updated at each step of the algorithm.
- y_{d_k,c_j}: represents the *actual* membership of the kth target word. In other words, this is the label provided in the supervised classification scheme. If c_j is the class of the kth target word, then $y_{d_k,c_j} = 1$; otherwise, $y_{d_k,c_j} = 0$.
- ϕ_{d_k,c_j}: represents the *obtained* membership of the kth target word. If c_j is the class obtained for the kth target word, then $\phi_{d_k,c_j} = 1$; otherwise, $\phi_{d_k,c_j} = 0$.
- ϵ_{d_k,c_j}: denotes the error of the current iteration. It is computed as:

$$\epsilon_{d_k,c_j} = y_{d_k,c_j} - \phi_{d_k,c_j}. \tag{3}$$

As we shall show, this error is used to update weights in f so that, at each new iteration, the distance between y_{d_k,c_j} and ϕ_{d_k,c_j} decreases.

Note that, in the model illustrated in Fig. 1, we only consider the relationship between feature and target words. The algorithm can be divided into the three following major steps:

1. **Initialization**: there are three possible ways of initializing f, i.e. the vector weights of feature words. The most simple strategy is to initialize weights with zeros or random values. A more informed alternative initializes weights using the a priori likelihood of feature words co-occur with senses. This probability can be computed as

$$\Pr = P(f_i|d_k) = n_{f_i,d_k}/n_{d_k}, \tag{4}$$

where n_{f_i,d_k} is the number of times that the ith feature word appears in the context of the kth target word and n_{d_k} is the total number of occurrences of d_k. In our experiments, we report the best results obtained among these three alternatives.
2. **Error calculation**: In the error calculation step, firstly, the output vector for each target word ($\phi(d_k)$) is computed. This vector depends upon the presence of the feature word in the context (w_{d_k,t_i}) and its relevance for the class (f_{t_i,c_j}). Mathematically, the class computed at each new iteration is given by

$$C\left(\sum_{t_i \in \mathcal{T}} w_{d_k,t_i} f_{t_i,c_j}\right) = \begin{cases} 1, & \text{if } c_j = \arg\max\limits_{c_l \in \mathcal{C}} \left(\sum\limits_{t_i \in \mathcal{T}} w_{d_k,t_i} f_{t_i,c_j}\right). \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Table 1
List of words used to evaluate the word sense disambiguation algorithm. NS and NI denote the number of senses of the target word and the number of instances in the corpus, respectively. The dataset comprising word context and word senses was obtained from previous studies in WSD [17,24,25]. Prior to the application of the learning methods, stopwords and punctuation marks were removed from the original instances.

Target word	NS	NI
interest (noun)	6	2368
line (noun)	6	4146
serve (verb)	4	4378
hard (adjective)	3	4333

After updating the classes for each target word, the values of f_{t_i,c_j} are modified. This update is controlled by the correction rate η:

$$f_{t_i,c_j}^{(n+1)} = f_{t_i,c_j}^{(n)} + \eta \sum_{d_k \in \mathcal{D}} w_{d_k,t_i} \epsilon_{d_k,c_j}^{(n)}, \tag{6}$$

where the superscript (n) in f and ϵ denotes the value of these quantities computed in the nth iteration of the algorithm and $\mathcal{D}$ is the set of target words. Note that $\epsilon_{d_k,c_j}^{(n)}$ is computed as defined in Eq. (3). The process of generating an output vector for each target word, computing the class and performing weight/feature relevance correction is done iteratively until a stop criterion is reached. In our experiments, we have stopped the algorithm when a minimum error $\epsilon_{\text{min}} = 0.01$ is obtained. If the minimum error is not reached after $n_{\text{max}} = 1,000$ iterations, the algorithm is stopped.

3. **Classification**: in the classification phase, the induced values of f are used in the classification. The word senses for each ambiguous word of the dataset are then obtained by computing the following linear combination:

$$\text{class}(d_k) = \arg\max_{c_j \in \mathcal{C}} \left(\sum_{t_i \in \mathcal{T}} w_{d_k,t_i} f_{t_i,c_j} \right). \tag{7}$$

Some aspects of the IMBHN algorithm resemble a neural network, namely the use of weights to represent the relevance of features in the classification process and the use of a similar optimization strategy to learn weights. However, the underlying IMBHN network structure completely differs from a neural network because the former learns a bipartite structure to represent the relationship of two distinct types of entities (terms and senses). Another distinctive difference of the IMBHN structure concerns its ability to direct propagate the information through neighbors. Note that a single layer network, conversely, performs the selection of relevant information via activation functions.

4. Experimental evaluation

The experimental evaluation of the algorithm was performed in two stages. In the first step, we assessed the performance of the algorithm by comparing to other state-of-the-art inductive classification algorithms. In the second stage, the IMBHN algorithm was applied to two WSD corpora previously used in WSD shared tasks, allowing thus the comparison of our method with state-of-the-art WSD systems. Both corpora are presented in the next section.

4.1. Corpora

4.1.1. Minimal corpus

The minimal corpus is composed of 4 words (*interest, line, serve* and *hard*), which were used in similar works [17,24,25]. This corpus comprises documents from distinct sources, including the San Jose Mercury News Corpus and the Penntreebank portion of the Wall Street Journal. The corpus encompasses 15,225 instances of short texts representing the context surrounding ambiguous words, where words are tagged with their respective part-of-speech. In this corpus, the correct senses conveyed by ambiguous words were manually annotated. The number of senses and the number of instances of each word used in our experiments is shown in Table 1. In the evaluation process, these four words were considered as the target words. In particular, to characterize the contexts, we have removed stopwords and punctuation marks as such elements do not convey any semantical meaning and, therefore, do not improve the characterization of contexts.

4.1.2. Senseval-3 and SemEval-2007

The Senseval-3 and SemEval-2007 corpora here presente refer, respectively, to the corpora used in the Senseval-3 English Lexical Sample Task [34] and in the SemEval-2007 Task 17 English Lexical Sample [40]. Both datasets provide instances of short texts representing the context of ambiguous words. The Senseval-3 is composed of 57 ambiguous words and 11,804 instances (7,860 for train and 3,944 for test). The words were extracted from the British National Corpus. The SemEval-2007 comprises 100 ambiguous words in 27,132 instances (22,281 for train and 4,851 for test). The data used in this corpus was extracted from both the Wall Street Journal and the Brown Corpus.

Table 2
Accuracy rates (%) obtained by each algorithm using *topical* features to disambiguate words. The studied target words are: (i) "interest" (noun), (ii) "line" (noun), (iii) "serve" (verb) and (iv) "hard". The best results for each value of $|\mathcal{T}|$ and for each target word are highlighted in bold font. The best results tend to occur with the SMO method, however, in particular cases, the J48 outperforms the SMO learning technique. Apart from the word "serve" when $|\mathcal{T}| = 300$, the IMBHN does not perform as good as the other traditional methods.

Method	$\|\mathcal{T}\|$	interest	line	serve	hard
IMBHN	100	71.49 (±1.90)	59.91 (±3.27)	64.68 (±3.63)	77.28 (±2.59)
J48	100	79.47 (±2.66)	62.73 (±1.94)	**68.15** (±1.34)	**84.58** (±1.29)
IBk	100	75.71 (±1.82)	53.18 (±2.36)	63.68 (±1.33)	79.34 (±2.29)
NB	100	59.79 (±2.56)	51.95 (±2.48)	58.79 (±1.84)	43.04 (±2.58)
SMO	100	**79.77** (±2.71)	**62.87** (±1.29)	66.79 (±1.21)	84.07 (±1.19)
IMBHN	200	78.50 (±2.61)	65.53 (±1.83)	66.56 (±2.43)	78.74 (±2.31)
J48	200	82.39 (±2.34)	66.71 (±2.22)	68.95 (±1.80)	**86.17** (±0.89)
IBk	200	80.70 (±2.10)	53.93 (±2.58)	63.24 (±2.47)	80.10 (±1.52)
NB	200	60.17 (±2.24)	54.43 (±2.92)	61.71 (±2.47)	42.69 (±2.62)
SMO	200	**83.27** (±2.51)	**68.95** (±1.72)	**69.84** (±1.70)	85.36 (±1.03)
IMBHN	300	80.23 (±2.31)	67.82 (±1.93)	71.42 (±1.55)	78.62 (±2.82)
J48	300	82.68 (±2.27)	68.54 (±1.26)	70.67 (±1.78)	**86.22** (±0.95)
IBk	300	80.32 (±2.14)	54.05 (±2.58)	63.13 (±2.29)	80.38 (±1.94)
NB	300	55.66 (±2.92)	54.14 (±2.61)	66.99 (±2.87)	41.61 (±2.49)
SMO	300	**84.71** (±1.93)	**69.87** (±0.87)	**71.92** (±2.25)	85.52 (±1.37)
Baseline	–	52.80	53.40	41.40	79.30

4.2. Experiment 1

In this experiment the results obtained by the IMBHN algorithm were compared with four inductive classification algorithms: Naive Bayes (NB) [18], J48 (C4.5 algorithm) [41], IB*k* (*k*-Nearest Neighbors) [1] and Support Vector Machine via sequential minimal optimization (SMO) [39]. The parameters of these algorithms have been chosen using the methodology described in [8]. For the IMBHN algorithm, we used the error correction rates $\eta = \{0.01, 0.05, 0.10, 0.50\}$. The number of topical features used in the experiments were $|\mathcal{T}| = \{100, 200, 300\}$. Finally, the window size for the local features were $\omega = \{1, 2, 3\}$. The evaluation process was performed via 10-fold cross-validation [23].

To analyze the behavior and accuracy of the IMBHN algorithm, we first studied the WSD task using topical features to characterize the context of target words of our dataset. The obtained results are shown in Table 2. When the number of topical features $|\mathcal{T}|$ is set with $|\mathcal{T}| = 100$, the best results occurred for the SMO and J48 techniques. In three cases, the IMBHN performed worse than the best results achieved with competing techniques.

In general, the performance of the classifiers tend to improve when the number of topical features ($|\mathcal{T}|$) increases from 100 to 300. This is clear when one observes that e.g. the best accuracy rate for the word "interest" goes from 79.77% to 84.71%. The same behavior can be observed for the other target words of the dataset, however, in a minor proportion. Concerning the performance of the IMBHN technique when $|\mathcal{T}| = \{200, 300\}$, in most cases, the IMBHN method is outperformed by the SMO technique, which provided the best results for the words "interest", "line" and "serve". The best results for the word "hard" was achieved with the J48 classifier.

When analyzing the performance of the classifiers induced with local features, a different pattern of accuracy has been found, as shown in Table 3. For the words "interest", "line"and "serve" the IMBHN classifier yielded the best results, for $\omega = \{1, 2, 3\}$. Conversely, if we consider the word "hard", the decision tree based algorithm, J48, outperformed all other methods. However, the performance achieved with J48 was very similar to the one obtained with the IMBHN: the maximum difference of accuracy between these two classifiers was 1.09%, when $\omega = 3$. This observation confirms the suitability of the method for the problem, as optimized results have been found for virtually all words of the dataset.

The best results obtained with topical and local features are summarized in Table 4. The IMBHN algorithm for representing texts and discriminating senses outperformed other methods when considering also distinct types of features. In special, the IMBHN performed significantly better than the SMO method for the word "line" and "serve". A minor gain in performance has been observed for "interest". With regard to the word "hard", the best performance was obtained with the J48 (with topical features). However, a similar accuracy was obtained with the IMBHN (with local features, as shown in Table 3). All in all, these results show, as a proof of principle, that the proposed algorithm may be useful to the word sense disambiguation problem, as optimal or near-optimal performance has been found in the studied corpus. Given the superiority of the local feature strategy, we also provide in Table S2 of the Supplementary Information results for additional words, which also confirm the effectiveness of the IMBHN algorithm.

State of the art WSD methods do not only use machine learning for classification purposes, but also a combination of heuristics, domain specific information and deep resources such as thesaurus and lexical datasets (e.g. the WordNet) [37]. The combination of distinct techniques and resources explains the reason why the IMBHN appears in a intermediary rank when compared to other methods relying upon more semantic information. We should note that the only information used

Table 3
Accuracy rates (%) obtained by each algorithm using *local* features to disambiguate words. The studied target words are: (i) "interest" (noun), (ii) "line" (noun), (iii) "serve" (verb) and (iv) "hard". The best results for each value of ω and for each target word are highlighted in bold font. For the words "interest", "line" and "serve", the best performance is achieved with the IMBHN method in all of the studied scenarios. For the word "hard", the J48 learning algorithm displayed the best performance. However, in this case, the IMBHN method performed almost as well as the J48, for $\omega = \{1, 2, 3\}$. Another interesting pattern arising from the results is the fact that performances are improved when ω takes higher values.

Method	ω	interest	line	serve	hard
IMBHN	1	**81.50** (±2.17)	**69.19** (±2.57)	**69.96** (±1.85)	85.50 (±1.46)
J48	1	65.83 (±2.86)	60.97 (±2.44)	46.43 (±2.54)	**85.57** (±1.02)
IBk	1	74.73 (±2.45)	59.76 (±2.39)	62.54 (±3.06)	82.06 (±1.82)
NB	1	64.90 (±3.63)	37.16 (±1.76)	42.11 (±2.20)	43.94 (±3.35)
SMO	1	66.00 (±2.33)	62.61 (±2.41)	57.88 (±2.73)	81.30 (±1.14)
IMBHN	2	**83.27** (±1.16)	**75.80** (±2.39)	**78.48** (±1.30)	84.67 (±1.64)
J48	2	71.74 (±2.01)	61.21 (±2.32)	55.57 (±2.67)	**85.39** (±1.03)
IBk	2	65.32 (±2.03)	56.72 (±2.70)	58.26 (±2.32)	78.35 (±1.21)
NB	2	66.97 (±1.83)	45.22 (±2.02)	60.16 (±2.87)	43.68 (±2.39)
SMO	2	64.10 (±2.65)	62.13 (±2.60)	58.63 (±3.74)	80.68 (±1.53)
IMBHN	3	**85.55** (±2.60)	**77.13** (±1.47)	**80.12** (±1.30)	84.16 (±0.65)
J48	3	76.85 (±2.75)	62.66 (±2.11)	60.94 (±2.41)	**85.25** (±1.08)
IBk	3	52.44 (±5.65)	53.59 (±2.27)	52.12 (±3.04)	78.86 (±1.17)
NB	3	68.49 (±1.92)	50.43 (±2.58)	66.05 (±2.03)	42.97 (±3.46)
SMO	3	64.14 (±2.35)	60.80 (±2.46)	58.45 (±3.24)	79.78 (±1.21)
Baseline	–	52.80	53.40	41.40	79.30

Table 4
Best classifiers for each feature set and its accuracy.

Target word	Topical features	Local features
interest (noun)	84.71% (SMO)	85.55% (IMBHN)
line (noun)	69.87% (SMO)	77.13% (IMBHN)
serve (verb)	71.92% (SMO)	80.12% (IMBHN)
hard (adjective)	86.22% (J48)	85.57% (J48)

by this method is the co-occurrence information present in the text, therefore no external information is used. Given the superiority of the IMBHN over SMO in some scenarios, it could be interesting to explore, in future works, the performance of other state of the art systems (such as the IMS) by using the IMBHN as the main machine learning algorithm (note that the IMS originally uses the SVM as main machine learning method).

A disadvantage associated to the use of supervised methods to undertake the word sense disambiguation problem is the painstaking, time-consuming effort required to build reliable datasets [37]. For this reason, it becomes relevant to analyze the performance of WSD systems when only a few labelled instances are available for training [37]. In this sense, we performed a robustness analysis of the proposed algorithm to investigate how performance is affected when smaller fractions of the dataset are provided for the algorithm. To perform such a robustness analysis the following procedure was adopted. We defined a sampling rate $\mathcal{S}$, representing the percentage of *disregarded* instances from the original dataset. For each sampling rate, we computed the accuracy $\Gamma(S)$ relative to the sampled dataset. The relative accuracy rate for a given S was computed as

$$\tilde{\Gamma}(S) = \frac{\Gamma(S)}{\Gamma(0)}, \tag{8}$$

which quantities the percentage of the original accuracy which is preserved when the original dataset is sampled with sampling rate S. For each sampling rate, we generated 50 sampled subsets. The obtained results for the IMBHN in its best configuration (i.e. using local features and $\omega = 3$) are shown in Fig. 2. The best scenario occurs for the word "hard", as even when 90% of the original is ignored, in average, more than 95% of the original accuracy (i.e. $\Gamma(S = 0)$) is recovered. Concerning the other words, a good performance was also observed when only a small fraction was available. This is the case of "serve": when 90% of the dataset is disregarded, 85% of the original accuracy is kept. These results suggest that the IMBHN could be successfully applied in much smaller datasets without a significative loss in performance. We have found similar robustness results for other configurations of parameters (ω) of the IMBHN (results not shown), which reinforces the hypothesis that the resiliency of the method with regard to the total amount of instances in the training phase is stable with varying parameter values. Note that such a robustness, although strongly desired in practical problems, does not naturally arise in all pattern recognition methods. This is evident e.g. when the robustness SMO is verified for "serve" and "interest", as shown in Fig. 3. Note that when $S = 0.9$, the accuracy drops to about 60% of its original value. The results confirmed that

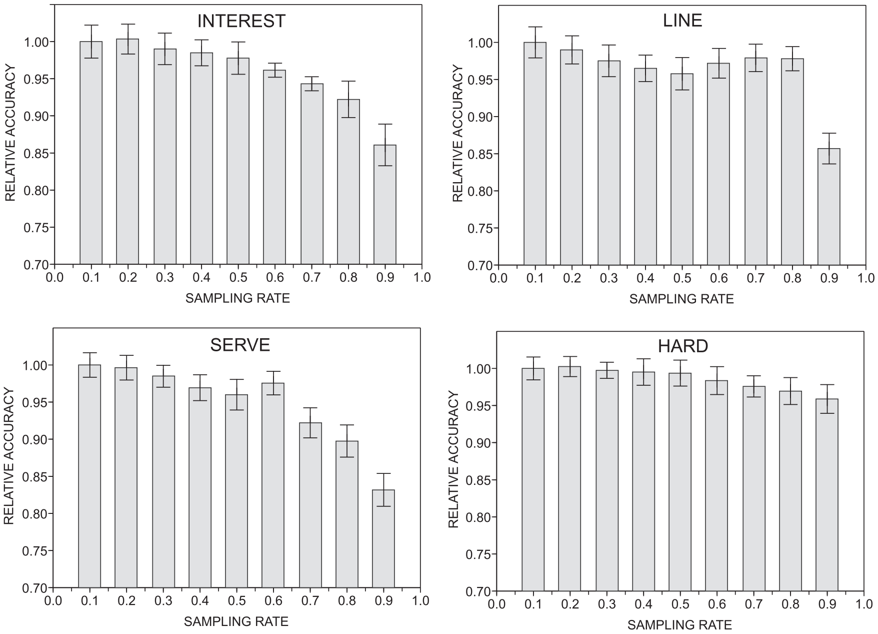

Fig. 2. Robustness analysis performed with the IMBHN algorithm. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by Eq. (8). Note that, in the worst case, the accuracy of the IMBHN reaches 85% of the accuracy when only 10% of the original data is available ($S = 0.9$), confirming thus the robustness of the method. A similar behavior was obtained when the approach based on topical features was evaluated with $\omega = \{1, 2\}$.

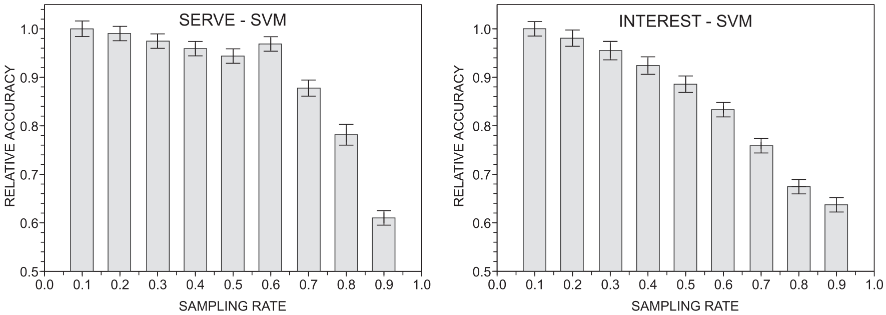

Fig. 3. Robustness analysis performed with the SMO algorithm for two words of the dataset. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by Eq. (8). Unlike the IMBHN algorithm, the accuracy rate drops significantly for high sampling rates.

only a minor decrease in performance is observed when labelled data is scarce in the IMBHN algorithm. Such a robustness suggests that the algorithm might not only be useful for the WSD task, but also for semi-supervised related problems [12].

Other important feature of any classifier are related to their scalability and time performance in the context of large instance problems [2,48]. The scalability issue of machine learning methods is oftentimes associated with two main aspects: the time required for (i) training and (ii) inference. According to Table S1 of the Supplementary Information, the IMBHN time performance is competitive when compared to other algorithms. Note that the internal operations can be performed in a matrix form, thus allowing an implementation based on specific efficient hardware, such as graphical processing units. Concerning the inference time, the IMBHN is also competitive compared to other methods, given that the state-of-art algorithms (such as IMS) rely on a SVM algorithm and therefore are much more less scalable with regarding to time performance.

In the proposed model, as the number of training examples increases, the connectivity patterns between feature and target words tend to become constant (i.e. each word tends to keep the same number of links). However, the number of links for each feature/target word depends on the ambiguous word being analyzed, so there is no simple clear pattern that can be explained with the degree of the bipartite networks. The same idea holds for other measurements such as those dependent on link weights. We note that topological features of networks, however, have already been used for the WSD task, with different network formations (see e.g. [9]). So we think that it would be interesting to explore in future works if there is any fact of the solution of the WSD problems that can be explained with features of bipartite networks.

Table 5
F-score obtained by the best result of the IMBHN and a sample of the systems that participated in the Senseval-3 along with the baseline (More Frequent Sense). The rank of each systems is based in its performance in fine coarse word sense disambiguation. Our system exceeded the baseline by 8.4% (fine) and 4.4% (coarse) besides having close results to the systems that were in 25th and 26th places.

Rank	System	Fine	Coarse
1	htsa3	72.9%	79.3%
25	UNED	64.1%	72.2%
–	**IMBHN**	63.6%	68.9%
26	SyntaLex-4	63.3%	71.1%
47	DLSI-UA-LS-NOSU	14.7%	23.9%
	Baseline(MFS)	55.2%	64.5%

Table 6
F-score obtained by the best result of the IMBHN and a sample of the systems that participated in the SemEval-2007 along with the baseline (More Frequent Sense). Our system exceeded the baseline by 5.2% and had close results to the systems that were in 6th and 7th places.

Rank	System	F-score
1	NUS-ML	88.7%
6	OE	83.8%
–	**IMBHN**	83.2%
7	VUTBR	80.3%
13	Tor	52.1%
	Baseline(MFS)	78.0%

4.3. Experiment 2

In this experiment, the IMBHN algorithm was applied in two WSD corpora that were previously used in Senseval-3 and SemEval-2007, allowing thus the comparison with state-of-the-art WSD systems that participated of the shared tasks. Only local features are considered in this experiment because, in the previous experiment, the best results of our method were obtained with these features. Since in both shared tasks the evaluation of WSD systems was performed using recall, precision and F-score, we chose to use the F-score because it consolidates recall and precision in a single quality index, simplifying the comparison between systems. The parameters of the algorithm were chosen in accordance with the previous experiment, being the error correction rate $\eta = 0.10$ and the window size for the local features $\omega = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We also considered as the context all words in the sentence where the ambiguous word occurs.

Table 5 shows the best result obtained by variations of the IMBHN, together with the baseline (i.e. the Most Frequent Sense) and a sample of the systems that participated in Senseval-3. The systems were evaluated in two variants, which considered fine and coarse grained senses (according to WordNet 1.7.1 and Wordsmyth). The ranking os systems was generated considering only their performance in fine grained senses. In this assessment, our system exceeded the baseline by 8.4% (fine) and 4.4% (coarse), and had very close results to the systems that were in 25th and 26th places (among 48 systems).

In the SemEval-2007 task, only coarse grained senses were considered (based on WordNet 2.1), since the identification of fine grained senses is a hard task even for human annotators [40]. Table 6 shows the result of the best variation of the IMBHN along with a sample of systems that participated in the SemEval-2007. We also show the performance of the baseline based on the most frequent sense. In this evaluation, our system outperformed the baseline by a margin of 5.2%. The IMBHN also displayed a similar performance of systems ranked in 6th and 7th places (among 14 systems).

In both datasets our algorithm did not exceed the best results, but managed to overcome the baseline and got better results than about half of the systems that participated in both tasks. Arguably, most of the participating systems have made the use of multiple features while we focused only statistical, superficial features. These results suggest that our system performs well if we consider that any linguistic, deeper information regarding senses was used to create the classifier. Another point of interest is that a large part of the best performing systems made use of the SVM as a core classifier. For this reason, we argue that such systems could benefit from the IMBHN to handle local features, since our algorithm is able to overcome the SVM in some cases, as discussed in the "Experiment1" section.

5. Conclusion

The accurate discrimination of word senses plays a pivotal role in information extraction and document classification tasks [4,14]. This task is important to improve other systems such as machine translators and search engines [37]. While methods based on deep paradigms may perform well in very specific domains, statistical methods based mainly on machine learning have proved useful to undertake the word sense disambiguation task in more general contexts. In this article, we have devised a statistical model to both represent contexts and recognize patterns in written texts. The model hinges on a bipartite network, with layers representing feature words and target words, i.e. words conveying two or more potential senses. We have shown, as a proof of principle, that the proposed model presents a significant performance, mainly when contextual features are modelled via extraction of local words to represent semantical contexts. We have also observed that, in general, our method performs well even if a relatively small amount of data is available for the training process. This is an important property as it may significantly reduce both time and effort required to construct a corpus of labelled data. Concerning its performance compared to state-of-the-art WSD systems, our method was competitive although not exceed the best methods that participated of the Senseval-3 English Lexical Sample Task and SemEval-2007 Task 17 English Lexical Sample. We note here that no deep linguistic information was used in our system, which makes it more suitable when the existence of such information is limited or absent. Even though our method does not present the lowest processing time, we highlight that the technique can take advantage of specific hardware, which may substantially improve the efficiency of the method in a practical scenario.

As future work, we intend to explore further generalizations of the algorithm. Owing to the power of word adjacency networks in extracting relevant semantical features of texts [9], we intend to use such models to improve the characterization of the studied bipartite networks. The word adjacency model could be used, for example, to better represent the relationship between feature and target words by using network similarity measurements [10,27,29]. We also intend to extend the present model to consider topological and dynamical measurements of word adjacency networks as local features [9]. While in the current model we explored only the relationship between feature words and target words, we could also consider the inner-relationships between feature words or target words. The relationship between features words, e.g. can be considered using other networked models, such as co-occurrence networks [44].

Acknowledgments

E.A.C. Jr. and D.R.A. acknowledge financial support from Google (Google Research Awards in Latin America grant). D.R.A. thanks São Paulo Research Foundation (FAPESP) for support (14/20830-0, 16/19069-9 and 17/13464-6). A.A.L. acknowledges support from FAPESP (11/22749-8 and 15/14228-9) and CNPq(Brazil) (302645/2015-2).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ins.2018.02.047.

References

[1] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[2] A. Alicante, M. Benerecetti, A. Corazza, S. Silvestri, A distributed architecture to integrate ontological knowledge into information extraction, Int. J. Grid Util. Comput. 7 (4) (2016) 245–256.
[3] D.R. Amancio, Authorship recognition via fluctuation analysis of network topology and word intermittency, J. Stat. Mech 2015 (3) (2015) P03005.
[4] D.R. Amancio, Comparing the topological properties of real and artificially generated scientific manuscripts, Scientometrics 105 (3) (2015) 1763–1779.
[5] D.R. Amancio, A complex network approach to stylometry, PLoS One 10 (8) (2015) e0136076.
[6] D.R. Amancio, Probing the topological properties of complex networks modeling short written texts, PLoS One 10 (2) (2015) e0118394.
[7] D.R. Amancio, E.G. Altmann, D. Rybski, O.N. Oliveira Jr, L.d.F. Costa, Probing the statistical properties of unknown texts: application to the voynich manuscript, PLoS One 8 (7) (2013) e67310.
[8] D.R. Amancio, C.H. Comin, D. Casanova, G. Travieso, O.M. Bruno, F.A. Rodrigues, L.d.F. Costa, A systematic comparison of supervised classifiers, PLoS One 9 (4) (2014) e94137.
[9] D.R. Amancio, O.N. Oliveira Jr, L.d.F. Costa, Unveiling the relationship between complex networks metrics and word senses, EPL (Europhys. Lett.) 98 (1) (2012) 18002.
[10] D.R. Amancio, O.N. Oliveira Jr., L.F. Costa, On the use of topological features and hierarchical characterization for disambiguating names in collaborative networks, EPL (Europhys. Lett.) 99 (4) (2012) 48002.
[11] D.R. Amancio, O.N. Oliveira Jr., L.F. Costa, Structure-semantics interplay in complex networks and its effects on the predictability of similarity in texts, Physica A 391 (18) (2012) 4406–4419.
[12] M.-F. Balcan, A. Blum, A discriminative model for semi-supervised learning, J. ACM 57 (3) (2010) 19:1–19:46.
[13] T. Berners-Lee, J. Hendler, O. Lassila, et al., The semantic web, Sci. Am. 284 (5) (2001) 28–37.
[14] A. Bouramoul, Contextualisation of information retrieval process and document ranking task in web search tools, Int. J. Space-Based Situated Comput. 6 (2) (2016) 74–89.
[15] F.A. 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 (9) (2012) 1686–1698.
[16] F.A. Breve, L. Zhao, M.G. Quiles, Semi-supervised learning from imperfect data through particle cooperation and competition, in: The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1–8.
[17] R. Bruce, J. Wiebe, Word-sense disambiguation using decomposable models, in: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, 1994, pp. 139–146.
[18] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, in: ICML '06, ACM, New York, NY, USA, 2006, pp. 161–168.
[19] K.W. Church, L.F. Rau, Commercial applications of natural language processing, Commun. ACM 38 (11) (1995) 71–79.
[20] G. Escudero, L. Màrquez, G. Rigau, J.G. Salgado, On the portability and tuning of supervised word sense disambiguation systems (2000).
[21] N. Fernandez, J.A. Fisteus, L. Sanchez, G. Lopez, Identity rank: named entity disambiguation in the news domain, Expert Syst. Appl. 39 (10) (2012) 9207–9221.
[22] D. Fernandez-Amoros, R. Heradio, Understanding the role of conceptual relations in word sense disambiguation, Expert Syst. Appl. 38 (8) (2011) 9506–9516.
[23] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.
[24] C. Leacock, G.A. Miller, M. Chodorow, Using corpus statistics and wordnet relations for sense identification, Comput. Linguist. 24 (1) (1998) 147–165.

[25] C. Leacock, G. Towell, E. Voorhees, Corpus-based statistical sense resolution, in: Proceedings of the workshop on Human Language Technology, Association for Computational Linguistics, 1993, pp. 260–265.

[26] Y.K. Lee, H.T. Ng, An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, Association for Computational Linguistics, 2002, pp. 41–48.

[27] E.A. Leicht, P. Holme, M.E.J. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2006) 026120.

[28] H. Liu, The complexity of chinese syntactic dependency networks, Physica A 387 (12) (2008) 3048–3058.

[29] J.-G. Liu, L. Hou, X. Pan, Q. Guo, T. Zhou, Stability of similarity measurements for bipartite networks, Sci. Rep 6 (2016) 18653EP.

[30] J.C. Mallery, Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers, Master's thesis, MIT Political Science Department, Citeseer, 1988.

[31] C.D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, USA, 1999.

[32] K. Markert, M. Nissim, Semeval-2007 task 08: Metonymy resolution at semeval-2007, in: Proceedings of the 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics, 2007, pp. 36–41.

[33] A.P. Masucci, G.J. Rodgers, Network properties of written human language, Phys. Rev. E 74 (2006) 026102.

[34] R. Mihalcea, T.A. Chklovski, A. Kilgarriff, The senseval-3 english lexical sample task, in: Proceedings of Senseval-3, Association for Computational Linguistics, 2004.

[35] R. Mihalcea, D. Radev, Graph-based Natural Language Processing and Information Retrieval, Cambridge University Press, 2011.

[36] R.J. Mooney, Comparative experiments on disambiguating word senses: an illustration of the role of bias in machine learning, arXiv:cmp-lg/9612001 (1996).

[37] R. Navigli, Word sense disambiguation: a survey, ACM Comput. Surveys 41 (2) (2009) 10.

[38] L. Pan, J. Cao, J. Hu, Synchronization for complex networks with markov switching via matrix measure approach, Appl. Math. Model 39 (18) (2015) 5636–5649.

[39] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.

[40] S.S. Pradhan, E. Loper, D. Dligach, M. Palmer, Semeval-2007 task 17: English lexical sample, srl and all words, in: Proceedings of the 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics, 2007, pp. 87–92.

[41] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[42] R.G. Rossi, A. de Andrade Lopes, T. de Paulo Faleiros, S.O. Rezende, Inductive model generation for text classification using a bipartite heterogeneous network, J. Comput. Sci. Technol. 29 (3) (2014) 361–375.

[43] S. Segarra, M. Eisen, A. Ribeiro, Authorship attribution through function word adjacency networks, IEEE Trans. Signal Process. 63 (20) (2015) 5464–5478.

[44] T.C. Silva, D.R. Amancio, Word sense disambiguation via high order of learning in complex networks, EPL (Europhys. Lett.) 98 (5) (2012) 58001.

[45] K. Sivaranjani, R. Rakkiyappan, J. Cao, A. Alsaedi, Synchronization of nonlinear singularly perturbed complex networks with uncertain inner coupling via event triggered control, Appl. Math. Comput. 311 (Supplement C) (2017) 283–299.

[46] K. Sneppen, M. Rosvall, A. Trusina, P. Minnhagen, A simple model for self-organization of bipartite networks, EPL (Europhys. Lett.) 67 (3) (2004) 349.

[47] D. Spina, J. Gonzalo, E. Amigó, Discovering filter keywords for company name disambiguation in twitter, Expert. Syst. Appl. 40 (12) (2013) 4986–5003.

[48] M. Steinbauer, G. Anderst-Kotsis, Dynamograph: extending the pregel paradigm for large-scale temporal graph processing, Int. J. Grid Util. Comput. 7 (2) (2016) 141–151.

[49] C. Stokoe, M.P. Oakes, J. Tait, Word sense disambiguation in information retrieval revisited, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, ACM, 2003, pp. 159–166.

[50] J. Véronis, Hyperlex: lexical cartography for information retrieval, Comput. Speech Lang. 18 (3) (2004) 223–252.

[51] G.A. Wachs-Lopes, P.S. Rodrigues, Analyzing natural human language from the point of view of dynamic of a complex network, Expert. Syst. Appl. 45 (2016) 8–22.

[52] W. Weaver, Translation, in: Machine Translation of Languages, 14, 1955, pp. 15–23.

[53] X. Yang, J. Cao, Hybrid adaptive and impulsive synchronization of uncertain complex networks with delays and general uncertain perturbations, Appl. Math. Comput. 227 (Supplement C) (2014) 480–493.

[54] O.N. Yaveroglu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, N. Przulj, Revealing the hidden language of complex networks, Sci. Rep. 4 (2014) 4547.

[55] Z. Zhong, H.T. Ng, It makes sense: A wide-coverage word sense disambiguation system for free text, in: Proceedings of the ACL 2010 System Demonstrations, Association for Computational Linguistics, 2010, pp. 78–83.

[56] X. Zhou, H. Han, Survey of word sense disambiguation approaches., in: FLAIRS Conference, 2005, pp. 307–313.