UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Using complex networks and Deep Learning to model and learn context

Edilson Anselmo Corrêa Júnior

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Edilson Anselmo Corrêa Júnior

Using complex networks and Deep Learning to model and learn context

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Diego Raphael Amancio

USP – São Carlos
February 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C825u

Corrêa Jr, Edilson Anselmo
Using complex networks and Deep Learning to model and learn context / Edilson Anselmo Corrêa Jr; orientador Diego Raphael Amancio. -- São Carlos, 2020.
92 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2020.

1. Context. 2. Complex Networks. 3. Deep Learning. 4. Ambiguity. I. Amancio, Diego Raphael, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Edilson Anselmo Corrêa Júnior

Modelagem e aprendizado de contexto usando redes complexas e Deep Learning

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Diego Raphael Amancio

USP – São Carlos
Fevereiro de 2021

ACKNOWLEDGEMENTS

I'd like to thank my advisor, Prof. Diego Amancio for his friendship, support, patience, guidance and dedication. I will definitely remember the many times we shared a cup of coffee, in which we had endless conversations and discussions.

I'd like to thank members of the NILC lab, some who became friends, especially Vanessa Q. Marinho and Leandro B. dos Santos. I would also like to thank colleagues from IFSC, Filipi N. Silva and Henrique F. de Arruda.

I'd like to thank my parents who over the years have demonstrated that support comes through actions rather than words. I would also like to thank my companion Andrine, for her support and understanding.

I'd like to thank the collaborators that I had over the years, I met wonderful people who allowed me to learn and grow.

Finally, I'd like to thank CAPES and the Google Research Awards for Latin America program for their financial support.

“The greatest enemy of knowledge is not ignorance,
it is the illusion of knowledge.”
(Daniel J. Boorstin)

ABSTRACT

CORRÊA JR, E. A. **Using complex networks and Deep Learning to model and learn context**. 2021. 92 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

The structure of language is strongly influenced by the context, whether it is the social setting, of discourse (spoken and written) or the context of words itself. This fact allowed the creation of several techniques of Natural Language Processing (NLP) that take advantage of this information to tackle a myriad of tasks, including machine translation, summarization and classification of texts. However, in most of these applications, the context has been approached only as a source of information and not as an element to be explored and modeled. In this thesis, we explore the context on a deeper level, bringing new representations and methodologies. Throughout the thesis, we considered context as an important element that must be modeled in order to better perform NLP tasks. We demonstrated how complex networks can be used both to represent and learn context information while performing word sense disambiguation. In addition, we proposed a context modeling approach that combines word embeddings and a network representation, this approach allowed the induction of senses in an unsupervised way using community detection methods. Using this representation we further explored its application in text classification, we expanded the approach to allow the extraction of text features based on the semantic flow, which were later used in a supervised classifier trained to discriminate texts by genre and publication date. The studies carried out in this thesis demonstrate that context modeling is important given the interdependence between language and context, and that it can bring benefits for different NLP tasks. The framework proposed, both for modeling and textual feature extraction can be further used to explore other aspects and mechanisms of language.

Keywords: Context, Complex Networks, Deep Learning, Ambiguity.

RESUMO

CORRÊA JR, E. A. **Modelagem e aprendizado de contexto usando redes complexas e Deep Learning**. 2021. 92 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

A estrutura da língua é fortemente influenciada pelo contexto, seja ele social, do discurso (falado e escrito) ou o próprio contexto de palavras. Este preceito propiciou a criação de várias técnicas de Processamento de Língua Natural (PLN) que tiram vantagem dessa informação para realizar uma miríade de tarefas, incluindo tradução automática, sumarização e classificação de textos. Entretanto, em grande parte dessas aplicações o contexto tem sido abordado apenas como uma informação de entrada e não como um elemento a ser explorado e modelado. Nesta tese, exploramos o contexto em um nível mais profundo, trazendo novas representações e metodologias. Ao longo da tese, consideramos o contexto como um elemento importante que deve ser modelado para melhor desempenhar as tarefas da PLN. Demonstramos como redes complexas podem ser usadas para representar e aprender informações de contexto durante a desambiguação do sentido das palavras. Além disso, propusemos uma abordagem de modelagem de contexto que combina *word embeddings* e uma representação de rede, esta abordagem permitiu a indução de sentidos de uma forma não supervisionada usando métodos de detecção de comunidade. Usando essa representação exploramos sua aplicação na classificação de textos, expandimos a abordagem para permitir a extração de características de texto com base no fluxo semântico, que foram posteriormente usadas em um classificador supervisionado treinado para discriminar textos por gênero e data de publicação. Os estudos realizados nesta tese demonstram que a modelagem de contexto é importante dada a interdependência entre linguagem e contexto, e que pode trazer benefícios para diferentes tarefas de PLN. O framework proposto, tanto para modelagem quanto para extração de características textuais, pode ser posteriormente utilizado para explorar outros aspectos e mecanismos da linguagem.

Palavras-chave: Contexto, Redes Complexas, Deep Learning, Ambiguidade.

CONTENTS

CHAPTER 1

INTRODUCTION

The study of language and how it is learned has been the focus of research in several areas of science, such as Linguistics, Psychology, Neuroscience and Computer Science (JURAFSKY; MARTIN, 2000). Given that these areas address the language from a specific perspective with different motivations and purposes, numerous discoveries and theories have been made over the years. However, the exact mechanisms, structures, processes and types of language learning are still considered unknown and they represent a rather old dispute between study approaches (for example, Rationalism versus Empiricism) (MARKIE, 2015). Nonetheless, some elements and concepts have been identified in more than one field as of great importance for understanding the workings of language, one of them being the *context*, which has its importance emphasized by the Firth's Contextual Theory of Meaning (FIRTH, 1957). This theory states that language and its structure is strongly influenced by the various forms of context: the context of the social environment, the context of spoken and written speech, the context of neighboring words among other contextual characteristics (MANNING; SCHUTZE, 1999).

In the daily use of language, references to context are common, expressions such as *'out of context'* or *'context of the situation'* are recurrent in cases where the interpretation of sentences, whole or partial texts and speeches, requires more information than is provided. This implies that context is informative about meaning (PIANTADOSI; TILY; GIBSON, 2012), that is, part of the meaning or interpretation attributed to a text depends partly on the context, with the degree of dependence varying. More important, considering language as an efficient communication system, redundancy on the information provided by the context will be sparsely and poorly provided or not provided at all, making ambiguity a prevalent phenomenon in language.

Another facet of context is that although is possible to define some of its influences on language the definition of context itself is ambiguous, it is possible to define context in very different ways depending on which language level (Morphology, Syntax, Semantics, Pragmatics) or textual element (paragraph, sentence and its components clauses, phrases, words or other structure) is being analyzed. Thus, it is necessary to define the context as an abstract concept that can represent different types of information, but which has a common element in relation to the language, which is to provide an additional layer of information, often nonexistent in the language itself.

In Artificial Intelligence (AI), more specifically, in Natural Language Processing (NLP), the importance of context is easily identified, methods and techniques make use of context information to perform a range of different tasks, such as in part-of-speech tagging (COLLOBERT; WESTON, 2008), summarization (RUSH; CHOPRA; WESTON, 2015), machine translation (SUTSKEVER; VINYALS; LE, 2014) and word sense disambiguation (JR; LOPES; AMANCIO, 2018). Despite the diversity of applications, the context used on these works is normally just the context of words (neighboring words). Instead of handling different contexts, it is common practice to create different systems, for example, when creating a sentiment classifier that will have as input texts from different sources (like Twitter and product review), different systems are created instead of dealing with the context, since the individual performance of specific sentiment classifiers is better than generic classifiers (CORRÊA *et al.*, 2017). So, it is possible to assert that in most works, context information have been neglected or poorly used, and hardly considered as an important part of the problem that needs to be modeled and explored along within the task.

Recently, two different fields have been exploring the importance of context in NLP, namely the areas of Complex Networks and Deep Learning. The application of the framework of complex networks to NLP is a trend that has been occurring in a similar way in several fields of science (COSTA *et al.*, 2011; BARONCHELLI *et al.*, 2013; CONG; LIU, 2014), in which problems modeled in graphs or networks are analyzed from a different and unexplored perspective not possible through traditional statistical techniques (COSTA *et al.*, 2007). When modeling text into networks, such as in co-occurrence networks (see details in Section A.3) that are highly dependent on the context of words, this type representation is not only able to capture information about the language and its structure but also about the context, explaining the success of its application in tasks such as text classification (ARRUDA; COSTA; AMANCIO, 2016), keyword extraction (MIHALCEA; TARAU, 2004) and word sense disambiguation (AMANCIO; JR; COSTA, 2012).

In Deep Learning, context has also been explored in many ways. One is in the representation of words in a vector space that is learned through the use of neural networks, the so-called *word embeddings* (BENGIO *et al.*, 2003). The Word2Vec model (MIKOLOV *et al.*, 2013a; MIKOLOV *et al.*, 2013b), one of the most popular models for generating *word embeddings*, uses the context where the words occur as the main source of information. Another example is the use of *attention* mechanisms in neural networks applied to machine translation. These mechanisms allow the system, when translating a word, to consider the importance of its neighboring words (context at word level) in the process (BAHDANAU; CHO; BENGIO, 2015). This mechanism significantly improved the performance of neural networks in this task (BAHDANAU; CHO;

BENGIO, 2015; LUONG; PHAM; MANNING, 2015). Another example the use of context is the Skip-Thought model, where embeddings are generated for entire sentences by using the surrounding sentences (context at sentence level) (KIROS *et al.*, 2015).

In both areas, Complex Networks and Deep Learning, the importance of *context* is visible, however, even though both use contextual information, they do it in very different ways, a fact that brings us to the focus of this thesis, which is the advancing of NLP models based on machine learning by combining the framework of complex networks and deep learning techniques in order to better model context in languages. In order to assess the effectiveness of the proposed context modeling methods we decide to explore two problems given all the possibilities in NLP, one being the ambiguity of words and the other text classification, more specifically genre and publication date classification.

Ambiguity is an inherent factor in the language and also a mechanism that allows the reuse of words, not forcing the human being to store an excessive number of different words (PIANTADOSI; TILY; GIBSON, 2012). Although important for the human being, ambiguity has a negative factor when it is necessary to automate some tasks related to language, such as automatic translation (NAVIGLI, 2009). This and other factors led to the emergence of the word sense disambiguation area, which aims to create automatic methods that can identify which sense of a word is being used in a given context. As is clear from the problem definition, this task has a high dependence on the context and a high complexity, being still an open problem in NLP and considered an IA-*complete* problem (analogous to NP-*complete* problems) (MALLERY, 1988; NAVIGLI, 2009), making it an ideal task to explore and apply the techniques that aim to model the context.

On the topic of text classification, it is possible to create machine learning models that completely disregard context or only consider word context (such as the bag of words representation) and still achieve good results. However, much about the textual structure and its creation process is left out, mostly because it is not necessary for simple text classification, taking this into account, we explore what information or features can be evidenced by the use of a modeling that considers context and whether these features can be used for more complex text classifications such as the discrimination of genre and publication date.

This work, as a thesis by articles or compendium, follows a structure where each chapter is a published article. So, in order to provide some insights on the period when the paper was written, we add two sections before each article, Context and Contributions. Another important aspect of this work is that all papers are self-contained in the regards of background information and methods, but the author also hopes that the reader, even without much background knowledge on complex networks and deep learning will be able to understand the papers and its contributions, for this reason we provide two appendices, presenting the basic principles and concepts of complex networks and deep learning. Also because of the chosen format for this thesis, the narrator will be referred to as “we,” rather than “I”, since the research was carried out in a collaborative setting.

The remainder of this thesis presents the articles, in Chapter 2 we explore if a complex network based machine learning model using only context information is capable of performing word sense disambiguation. In Chapter 3 we investigated the possibility of representing the context by combining word embeddings and network representation, we also evaluated the use of this methodology in the word sense induction task. In Chapter 4 we explore if the context modeling proposed could capture other types of information, more precisely, genre and temporality. Finally, at the end, we present a general conclusion.

CHAPTER 2

WORD SENSE DISAMBIGUATION: A COMPLEX NETWORK APPROACH

Word sense disambiguation: A complex network approach. Edilson A Corrêa Jr, Alneu A Lopes, Diego R Amancio. *Information Sciences 442*, 103-113, 2018.

2.1 Context

The state of the art approach in supervised word sense disambiguation at the time we wrote this article was usually a combination of heuristics, domain specific information and linguistic resources such as lexical datasets and thesaurus, topped by a supervised classification method. A good example is It Makes Sense (IMS) (ZHONG; NG, 2010), in its default configuration, IMS makes use of three set of features to characterize a ambiguous word, the POS tag of surrounding words, words in the context and local collocations [1], all these features are used as input for the Support Vector Machines (SVMs), a supervised classification algorithm.

In a parallel research path, some works, looking for alternative solutions to the WSD problem, explored the use of graphs (VÉRONIS, 2004; MIHALCEA; RADEV, 2011) and later complex networks (SILVA; AMANCIO, 2012; AMANCIO; JR; COSTA, 2012). Graphs and networks were both used as a framework for disambiguation algorithms but also as a representation that allowed the extraction of features (complex networks measurements), that had a different nature than the traditional features used in WSD systems. These methods didn't overcome the state of the art in WSD, but as demonstrated by Amancio, Jr and Costa (2012), the performance of traditional methods can be improved when combined with network-based features.

These two lines of work motivated us to explore a method that would provide an

[1] A collocation is a expression of two or more words that tend to appear frequently together, in which the whole is perceived to have a meaning beyond the sum of the parts (MANNING; SCHUTZE, 1999)

intersection between the two. A machine learning method that would perform the learning process in the same structure used for the representation (a complex network), instead of just extracting features.

2.2 Contributions

The contributions of this paper can be summarized in two topics. First, we brought an adaptation of the Inductive Model Based on Bipartite Heterogeneous Network (IMBHN) algorithm to the context of word sense disambiguation, the method proved to be suited for the task, having competitive performance in two WSD shared tasks (Senseval-3 English Lexical Sample Task and SemEval-2007 Task 17 English Lexical Sample). Although the method did not outperform the state of the art, in a separate analysis, we showed that when compared to traditional algorithms such as SVMs, the method had better results, indicating that systems such as IMS could benefit from the IMBHN to handle context features instead of SVMs.

Second, the proposed adaptation, instead of using features that were extracted from context, it explicitly modeled the context, demonstrating that context can be further explored to tackle NLP problems through complex network representations. This insight allowed us to follow this path and move to our other two works.

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} τ_ω 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

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

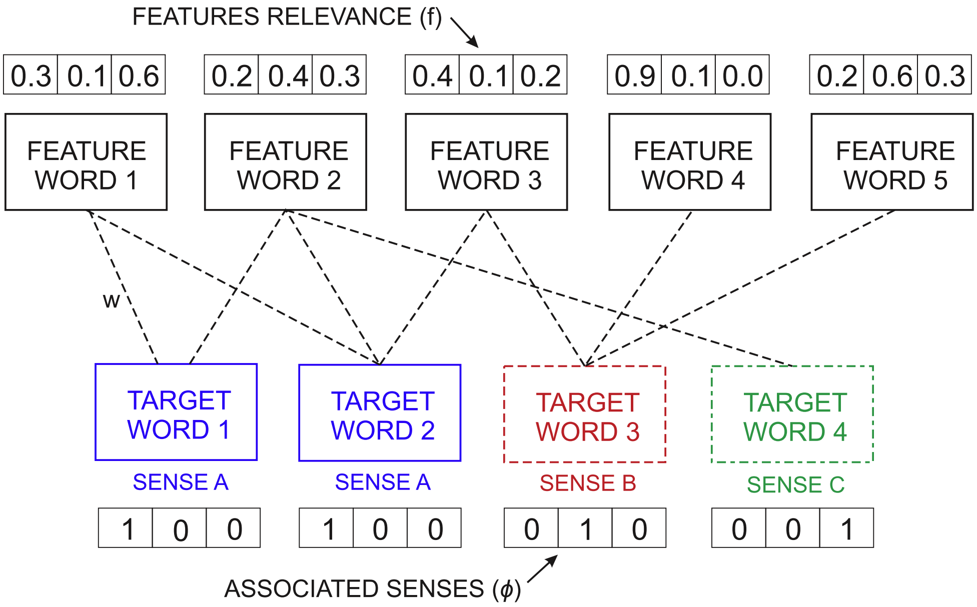


Fig. 1. Bipartite network structure used by the IMBHN algorithm. Note the existence of two layers: the layer comprising feature words and the layer comprising target words, which can be classified into three distinct senses (A, B and C). For each feature word, there exists a vector of features relevance 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:

$$\epsilon_{d_k,c_j} = y_{d_k,c_j} - \phi_{d_k,c_j}. \quad (3)$$

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

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

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

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

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

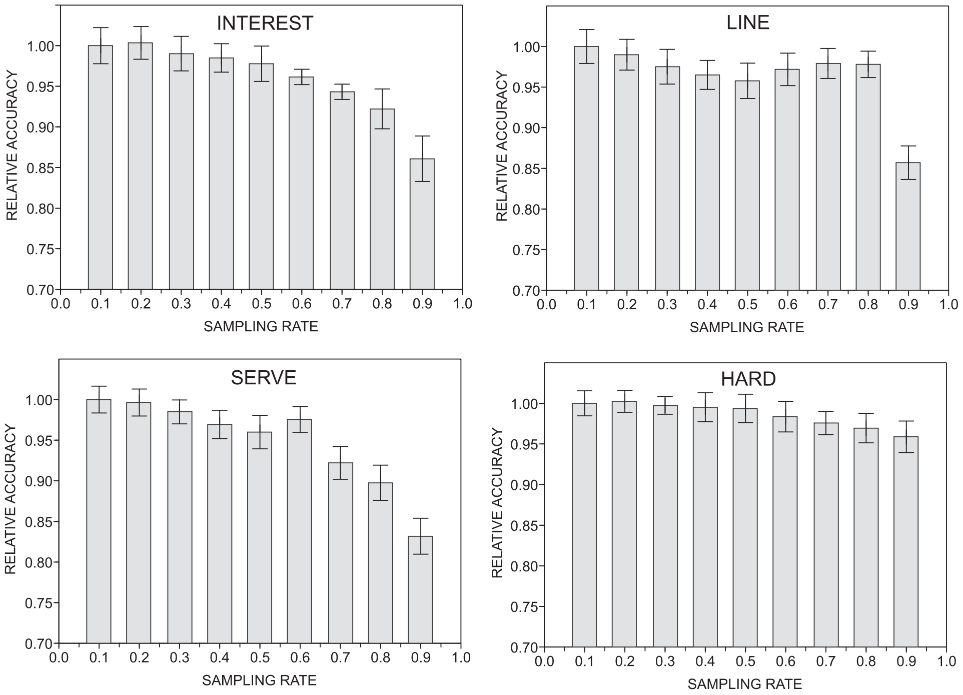


Fig. 2. Robustness analysis performed with the IMBHN algorithm. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by Eq. (8). Note that, in the worst case, the accuracy of the IMBHN reaches 85% of the accuracy 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 $\omega = \{1, 2\}$.

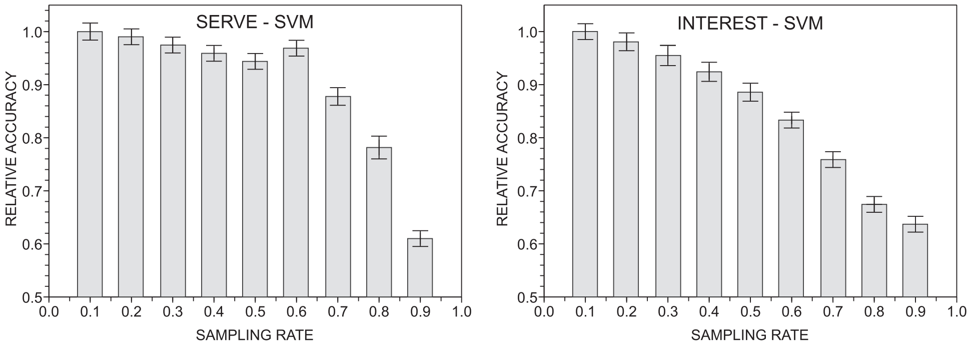


Fig. 3. Robustness analysis performed with the SMO algorithm for two words of the dataset. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by Eq. (8). Unlike the IMBHN algorithm, the accuracy rate drops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

CHAPTER 3

WORD SENSE INDUCTION USING WORD EMBEDDINGS AND COMMUNITY DETECTION IN COMPLEX NETWORKS

Word sense induction using word embeddings and community detection in complex networks. Edilson A Corrêa Jr, Diego R Amancio. *Physica A: Statistical Mechanics and its Applications 523*, 180-190, 2019.

3.1 Context

The importance of context in the field of NLP is not a recent discovery, works like Schütze (1992) have used context information in the task of disambiguation in the early days of the field, however, once more, its importance has been reinforced in our work (previous chapter). We demonstrated that a network representation that reinforces the importance of context together with a learning algorithm is capable of obtaining good results in the task of word sense disambiguation without using any other source of information. Further exploring context information in disambiguation tasks, other works made use of pre-trained word embeddings (see B.2) to represent context. Iacobacci, Pilehvar and Navigli (2016) combined pre-trained word embeddings of the words in context of an ambiguous word and used that combination as feature in the IMS framework, obtaining state of the art results. Kågebäck *et al.* (2015) not only combined pre-trained word embeddings, but proposed a combination method that uses specific weights for each word in context, this method combined with a traditional clustering method (k-means) achieved state of the art results in unsupervised disambiguation, more specifically, word sense induction.

At the same time that the use of word embeddings was being widespread in several NLP tasks, related studies were also being carried out in the area of complex networks. Perozzi *et al.*

(2014) showed that word embeddings could be represented in a network structure and that the resulting networks could bring important information about the represented embeddings, such as a meaningful community structure.

These works motivated us to represent the context of words in a network, which in turn is modeled through the combination of word embeddings. In addition to the representation, we also explore the use of community detection methods as an alternative to traditional clustering methods.

3.2 Contributions

The main contribution of this paper was to explore the concept of context embeddings modeled as complex networks and to use this structure to induce word senses via community detection algorithms. Once more, our system was based only on features of the context and was able to overcome competing algorithms and baselines. Being a method that does not use any additional information and is completely unsupervised, it makes possible its insertion in a fully automated system.

Going a step further, although the representation in this work has been used for the task of inducing senses, the proposed framework is generic enough to absorb any graph-based applications in scenarios where unsupervised methods are required to process natural languages.

3.3 Recent Developments

In addition to the use of word embeddings in the disambiguation process, both supervised and unsupervised, some works began to seek representations of words that assume that a word has more than one sense, thus creating multiple embeddings for a single word, describing the multiple meanings that a word may have (NEELAKANTAN *et al.*, 2014; IACOBACCI; PILEHVAR; NAVIGLI, 2015; PELEVINA *et al.*, 2016; SCARLINI; PASINI; NAVIGLI, 2020).

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/© 2019 Elsevier B.V. All rights reserved.

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

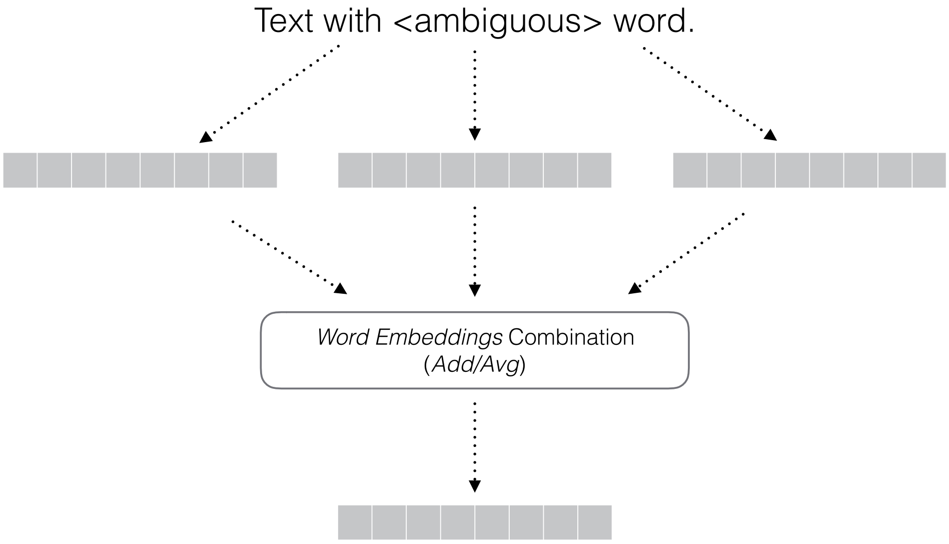


Fig. 1. Example illustrating how the context can be characterized from individual word embeddings. Given the word vectors representing the word appearing in the context, we combine those vectors to obtain a single embedding representing the context around the ambiguous word.

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].

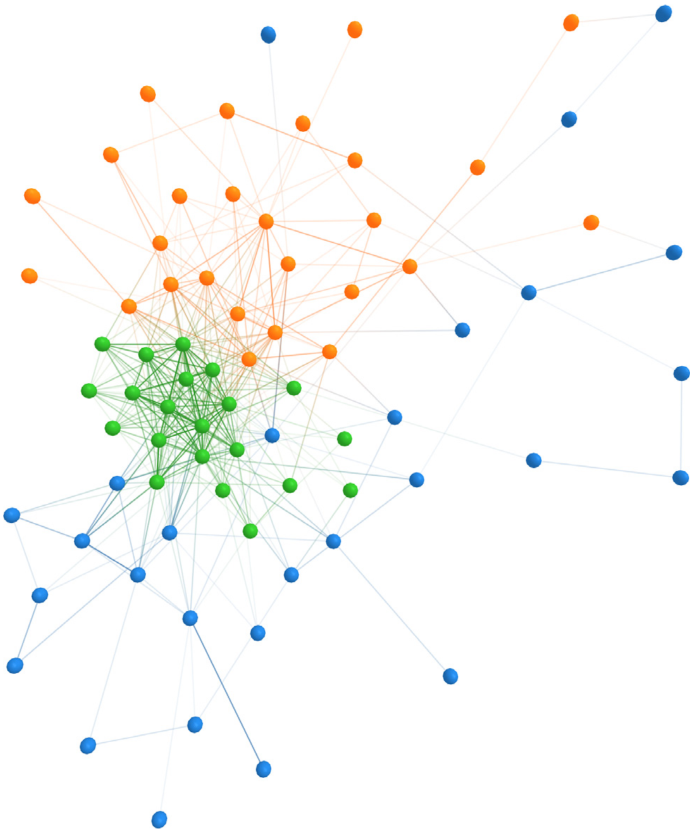


Fig. 2. Example of network obtained from the proposed model using $\omega = 10$ for the CN-ADD model. Each distinct color represents a different sense induced for the word "add". The visualization was obtained with the *networks3d* software [41].

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.

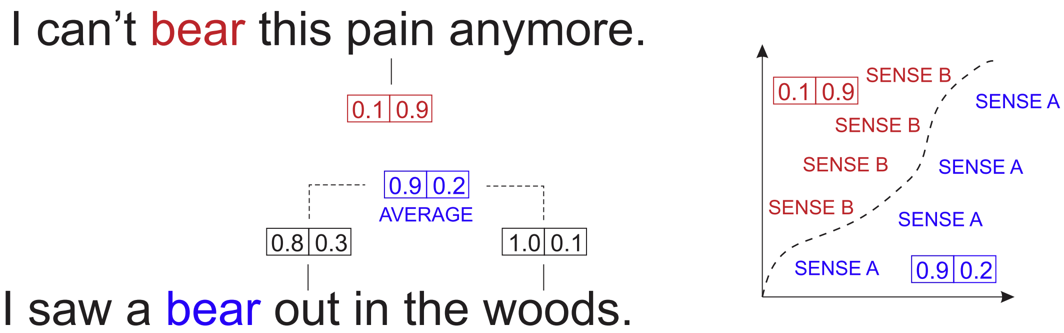


Fig. 3. Example of how senses are classified according to our methodology. In this example, the word "bear" can convey two different meanings. Note that the classification of senses relies on the embeddings of context words.

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.

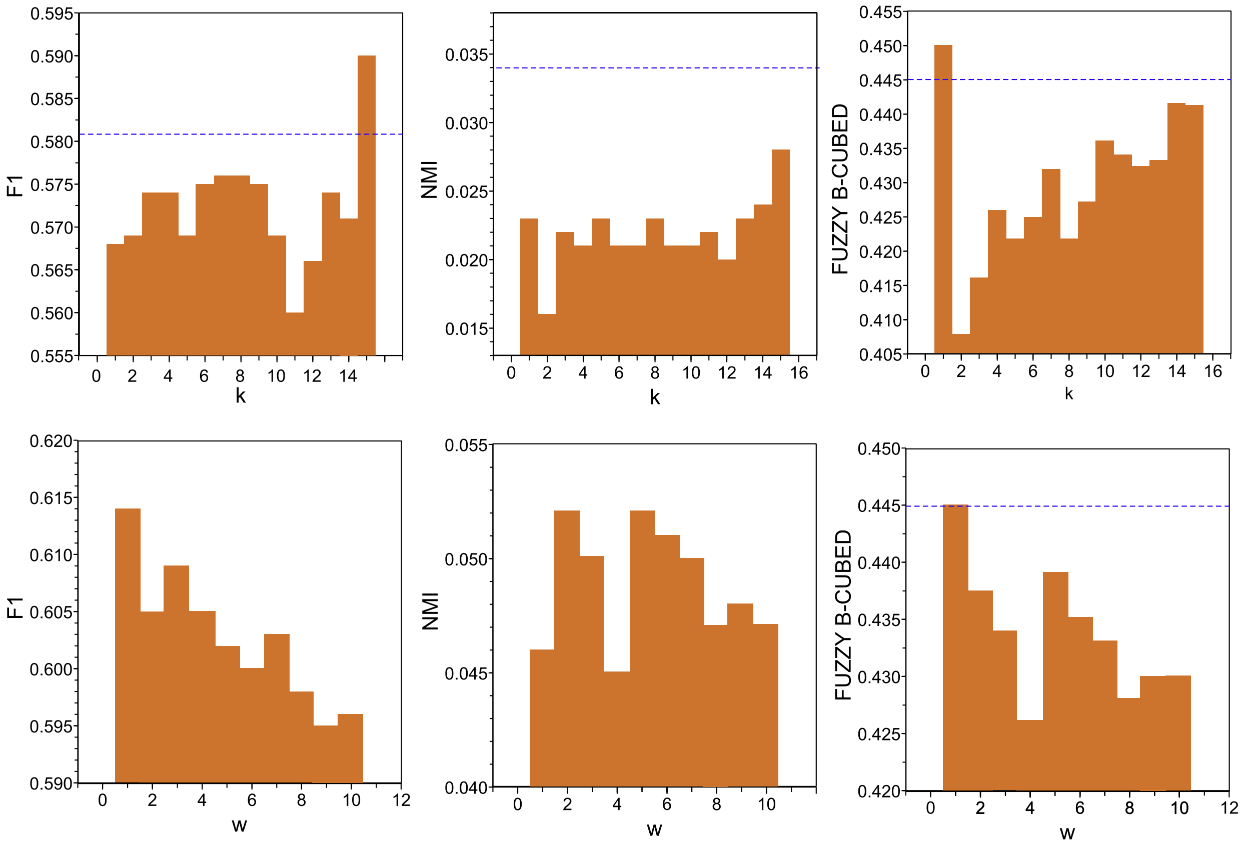


Fig. 4. Dependence of the performance results using different configuration of parameters. In all figures, we show the scenario allowing only one sense for the ambiguous word. In (a), (b) and (c); we analyze the behavior of the performance as a function of k when the full context of the sentence is considered. 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.

CHAPTER 4

SEMANTIC FLOW IN LANGUAGE NETWORKS DISCRIMINATES TEXTS BY GENRE AND PUBLICATION DATE

Semantic flow in language networks discriminates texts by genre and publication date. Edilson A Corrêa Jr, Vanessa Q Marinho, Diego R Amancio. *Physica A: Statistical Mechanics and its Applications 557*, 124895, 2020.

4.1 Context

When using the combination of complex networks and word embeddings, we were able to successfully represent context and induce senses from it. With this result, we questioned whether the context modeling we used could capture other types of information. Thus, using the same context modeling to model sentences from literary pieces, we investigate what information could be extracted and used in later tasks.

4.2 Contributions

The main contribution of this work was the adaptation of the framework used to represent contexts and to induce senses that was presented in our previous work, for the representation of texts. In addition to the adaptation, we also extended the framework so that it was able to capture and characterize the semantic flow of the text. This characterization allowed us to later discriminate texts by genre and also by date of publication.

4.3 Recent Developments

During and after the writing of this work, the application of complex networks to capture linguistic phenomena has expanded, one example is the study of the semantic and emotional structure of suicide notes (TEIXEIRA *et al.*, 2020). Another element that has become quite popular is the enrichment of the network structure by using word embeddings (SANTOS *et al.*, 2017; QUISPE; TOHALINO; AMANCIO, 2020).

Physica A 557 (2020) 124895

Contents lists available at ScienceDirect

Physica A

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

Semantic flow in language networks discriminates texts by genre and publication date

Edilson A. Corrêa Jr., Vanessa Q. Marinho, Diego R. Amancio [*]

Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 7 January 2020
Received in revised form 6 April 2020
Available online 30 June 2020

Keywords:
Complex networks
Semantic networks
Co-occurrence networks
Language networks
Word embeddings
Network embeddings

ABSTRACT

We propose a framework to characterize documents based on their semantic flow. The proposed framework encompasses a network-based model that connected sentences based on their semantic similarity. Semantic fields are detected using standard community detection methods. As the story unfolds, transitions between semantic fields are represented in Markov networks, which in turn are characterized via network motifs (subgraphs). Here we show that different book characteristics (such as genre and publication date) are discriminated by the adopted semantic flow representation. Remarkably, even without a systematic optimization of parameters, philosophy and investigative books were discriminated with an accuracy rate of 92.5%. While the objective of this study is not to create a text classification method, we believe that semantic flow features could be used in traditional network-based models of texts that capture only syntactical/stylistic information to improve the characterization of texts.

1. Introduction

In the last few years, several interesting findings have been reported by studies using network science to model language [1–10]. Network-based models have been used e.g. to address the authorship recognition problem, where the structure of the networks can provide valuable language-independent features. Other relevant applications relying on network science include the word sense disambiguation task [11,12], the analysis of text veracity and complexity [13,14]; and scientometric studies [15].

Whilst most of the network-based language research have been carried out at the word level [16,17], only a limited amount of studies have been performed based on mesoscopic structures (sentences or paragraphs) [18]. In addition, most of the studies have analyzed language networks in a static way [19,20]. In other words, once they are obtained, the order in which nodes (words, sentences, paragraphs) appear is disregarded. Here we probe the efficiency of sentence-based language networks in particular classification problems. Most importantly, differently from previous works hinging on network structure characterization [16,17], we investigate whether the semantic flow along the narrative is an important feature for textual characterization in the considered classification tasks.

During the construction of a textual narrative, oftentimes authors follow a structured flow of ideas (introduction, narrative unfolding and conclusion). Even in books displaying a non-linear, complex narrative unfolding, one expects that an underlying linear semantic flow exists in authors' mind. In other words, even though narrative events might not organize themselves in a trivial linear form, the linearity imposed by written texts requires some type of linearization (e.g. by performing a walk through the network). This idea is illustrated in Fig. 1.

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

https://doi.org/10.1016/j.physa.2020.124895

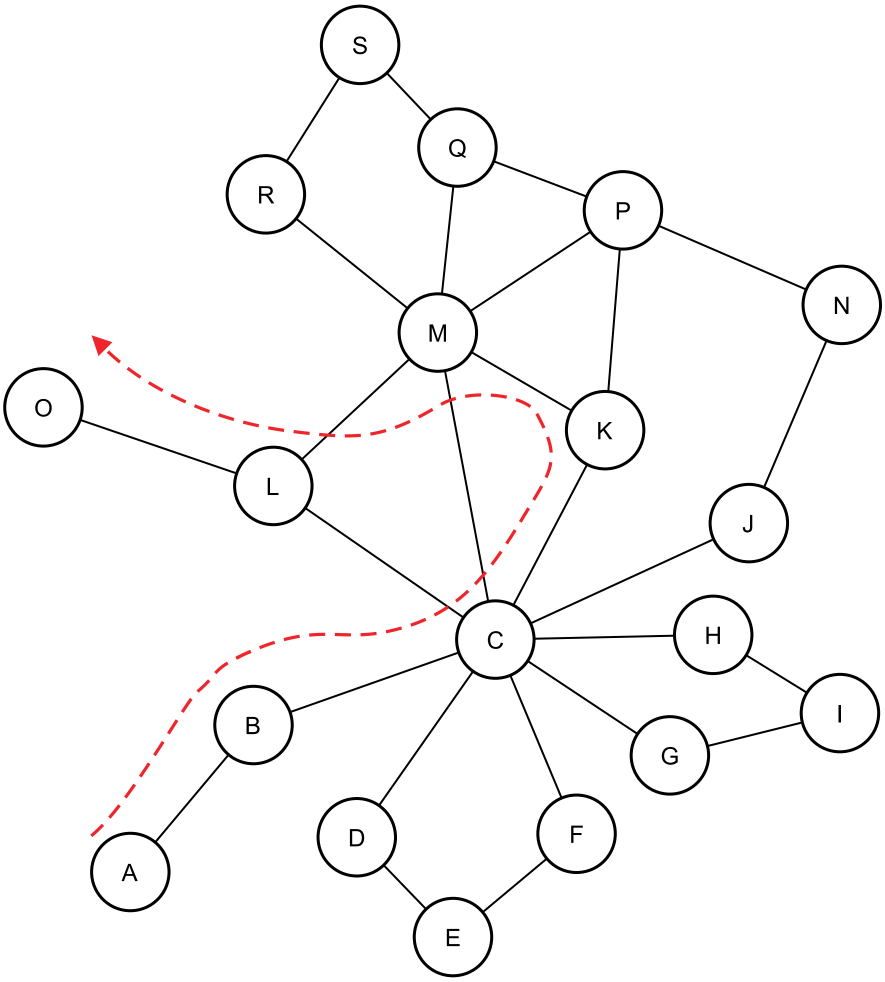


Fig. 1. High-level scheme illustrating the process of creating a text from a network of ideas. Each node represents an idea in the text. Usually, in semantic networks, nodes can represent words, sentences, paragraphs or even a sequence of paragraphs [12,18]. Edges represent the relationship (similarity) between ideas. In this paper we model each sentence as a distinct node in the network. A written text can be seen as a walk on this network (see e.g. [21]). In the example, the following sequence of ideas is produced: "A, B, C, K, M, L, O".

The ideas conveyed by a text can be represented as a complex network, where nodes represent semantic blocks (e.g. sentences, paragraphs), and edges are established according to semantic similarities. To map such a conceptual network into a text, authors perform a linearization process, where nodes (concepts, ideas) are linearly chosen and then transformed into a linear narrative (see Fig. 1). Such a projection of a multidimensional space of ideas into a linear representation has been object of studies both on network theory and language research. A consequence of such a linearization in texts is the presence of long-range correlations at several linguistic levels, a property that has been extensively explored along the last years [22–25].

While complex semantic networks have been used in previous works to represent the relationship between ideas and concepts, only a minor interest has been devoted to the analysis of how authors navigate the high-dimensional semantic relationships to generate a linear stream of words, sentences or paragraphs. In [26], a mesoscopic representation of networks was proposed. The authors used as a semantic, meaningful block a set of consecutive paragraphs. The semantic blocks were connected according to a lexical similarity index. The model aimed at combining a networked representation with an idea of semantic sequence obtained when reading a document. Even though some interesting patterns were found, the concept of semantic fields were not clear, as no semantic community structure arises from mesoscopic networks. The problem of linearization of a network structure was studied in [21]. A systematic analysis of the efficiency of several random walks in different topologies was probed. The efficiency was probed in a twofold manner: (i) the efficiency in transmitting the projected network; and (ii) the efficiency in recovering the original network. In [27], the authors explored the efficiency of navigating an idea space, by varying network topologies and exploration strategies.

In the current paper, we take the view that authors write documents by applying a linearization process to the original network of ideas, as shown in the procedure illustrated in Fig. 1. Upon analyzing the flow of ideas with the adopted network-based framework, we show that features extracted from the networks can be employed to characterize and classify texts. More specifically, we defined the network of ideas as a network of sentences linked by semantic similarity. *Semantic fields* of similar sentences (nodes) were identified via network community detection. These fields (network communities) were then used to characterize the dynamics of authors' choices in moving from field to field as the story unfolds. Using a stochastic Markov model to represent the dynamics of choices of semantic fields performed by the author along the text, we showed, as a proof of principle, that the adopted representation can retrieve textual features including style (publication epoch) and complexity.

2. Research questions

The main objective is to answer the following research questions: is there any patterns of semantic flow in stories? Are these patterns related to textual characteristics? To address these questions, we use sentence networks to represent the semantic flow of ideas in texts. Such networks are summarized using a high-level representation based on the relationship between communities extracted from the sentence networks. Using this representation, we show that motifs extracted from such a high-level representation can be used to classify texts according to the style in which authors unfolds their stories. We are not proposing a novel text classification method, but investigating whether semantic flow is a feature that depends on text genre and publication date. We argue that the obtained results suggest that the proposed high-level view of a text network could be further probed in other Natural Language Processing classification tasks.

3. Materials and methods

This study can be divided in two parts. In the first step, we identify the semantic clusters (fields) of the story. Differently from the analysis of short texts, where semantic groups can be identified mostly by identifying paragraphs, in long texts – the focus of this study – the identification of semantic clusters is more challenging because semantic topics might not be organized in consecutive sentences/paragraphs owing to the linearization process illustrated in Fig. 1. In other words, the process of obtaining semantic clusters can be understood as the reverse operation depicted in Fig. 1.

In order to identify semantic clusters from the text, we first create a network of sentences for each document, where sentences are linked if the similarity between them is above a given threshold. The obtained network is then analyzed via community detection methods, where groups of densely connected sentences are identified and considered as semantic clusters. A qualitative analysis of the obtained communities suggested that most of the largest communities are in fact related to a specific subtopic approached in the text. This idea relating semantic fields and network communities has also been used to construct automatic summarization systems [28].

In the second step of this study, we investigate the semantic flow of ideas developed by authors while unfolding their stories. We consider each community found as a semantic cluster, and as the story unfolds (one sentence after another), we analyze the community labels of the adjacent sentences to create a Markov chain, where each state represents a community and transitions are given by the text dynamics. Once the Markov chain representing the transitions of semantic clusters is obtained, the text is characterized by finding and counting different chain motifs. Such a characterization is then used to classify texts according to the semantic flow as revealed by sentences membership to different network communities.

The main objective of this work is to provide a framework to analyze and verify whether the semantic flow in texts can be used to characterize documents. Because the framework encompasses some steps, several alternatives could be probed in each step. We decided not to conduct a systematic analysis of combination of methods (and parameters) owing to the complexity of such analysis. A systematic study of the parameters and methods optimizing the proposed framework is intended to be conducted as a future work.

In Fig. 2 we show a representation of the framework proposed to analyze stories. In the next section, we detail each of steps used in this framework.

3.1. Word and sentence embeddings

Usually any vector representation of words is known as a *word embedding*. However, since the creation of *neural word embeddings* [29], the term is mostly used to name those approaches based on neural network representations. The *word embedding* model proposed in [29] aimed at classifying texts based on raw text input. Thus, the classification does not require that textual features as input. Typically, *word embeddings* are dense vectors that are learned for a specific vocabulary, with the objective of addressing some task.

A typical task addressed with word embeddings is the language modeling problem, which aims at learning a probability function describing the sequence of words in a language. More recently, this same vector representation has been used in more complex models, with the objective of addressing several Natural Language Processing tasks simultaneously, including POS tagging, name entity recognition, semantic role labeling and others [30,31]. Despite its relative success in the above mentioned tasks, the adopted embeddings could not be used in general purpose applications [30,31]. In order to allow the use of embeddings in wider contexts, the Word2Vec representation was proposed [32,33].

The Word2Vec is a neural model proposed to learn a dense, high-quality representation that is able to capture both syntactical and semantical language properties. As a consequence, vectors representing words conveying the same meaning are close in the considered space. An interesting property of the Word2Vec technique is the *compositionality*, which allows that large information blocks (e.g. sentences) can be represented by combining the representation of the

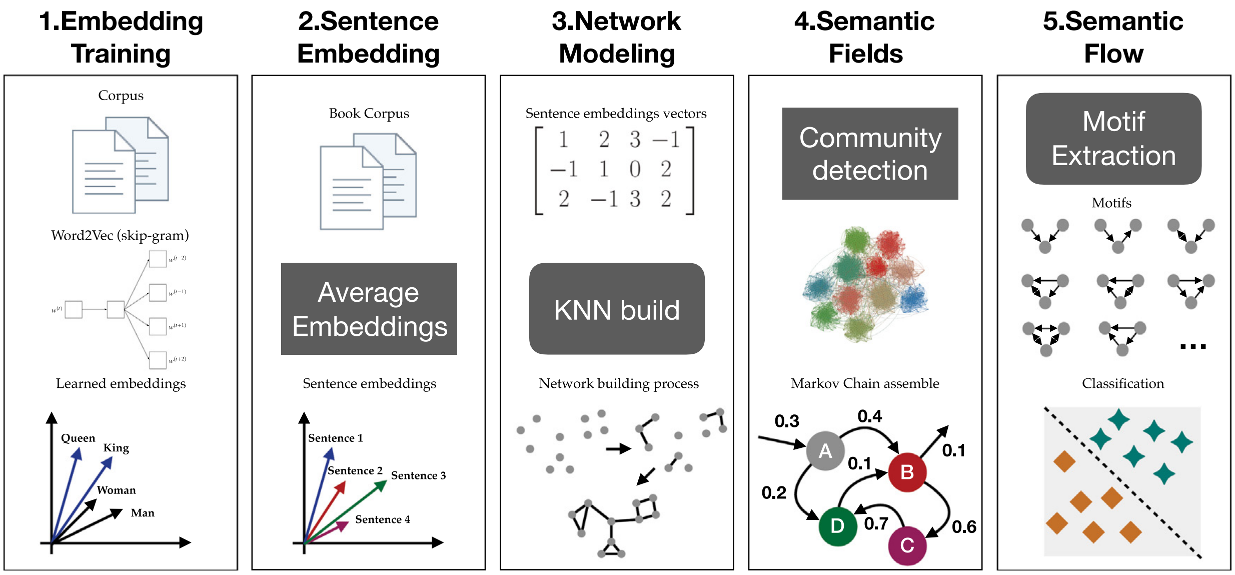


Fig. 2. Sequence of steps employed to characterize documents using the proposed framework: (1) word embedding generation; (2) sentence embeddings generation from word embeddings; (3) a sentence similarity network is generated based on the similarity of sentence embeddings; (4) network communities are detected and a Markov chain is built based on the story unfolding (semantic flow); and (5) motifs are identified in the Markov chain representing the semantic flow. These motifs are then used as features in a classification method.

vector representing the words in the sentence. Other interesting property is the ability to combine embeddings in a intuitive fashion [32,34]. For example, using the Word2Vec technique, the following relationship can be obtained:

$$\text{vector}(\text{“King”}) - \text{vector}(\text{“Man”}) + \text{vector}(\text{“Woman”}) \simeq \text{vector}(\text{“Queen”}). \tag{1}$$

The Word2Vec model is a robust, general-purpose neural representation that has been widely used in several Natural Language Processing tasks, including machine translation [35], summarization [28,36], sentiment analysis [37] and others. Given the success of the this model and the possibility of composition in different scenarios (sentiment analysis and sense disambiguation) [37,38], in the current study we used a representation of sentences based on the Word2Vec. We have chosen the Word2Vec as embedding method to illustrate the proposed framework for the aforementioned advantages. However, we note that other embeddings techniques exist [39,40]. A comparison of techniques, however, revealed no significant difference in performance.

More specifically, here the embedding **s** of a sentence s is represented by the average embedding of the words in s:

$$\mathbf{s} = \frac{1}{\omega(s)} \sum_{i=0}^{\omega} \mathbf{w}_i. \tag{2}$$

where $\mathbf{w}_i$ is the embedding of the ith word in s and $\omega(s)$ is the total number of words in s.

The word embedding technique used here was obtained with the Word2Vec method (skip-gram). The training phase used the Google News corpus [32,33]. According to [32,33], the parameters of the method are optimized in the context of semantical similarity task. The combination of embeddings to represent a sentence in Eq. (2) could also be performed by summing individual embeddings. However, it has been shown that there is no significant difference when sentence embeddings are used to construct a network of sentence similarity [41]. We note that some words are removed from this analysis. This includes *stopwords* (e.g. articles and prepositions) and words with no embeddings in the Google News corpus. Thus, whenever a sentence contains only words with no available embeddings, it is removed from the analysis.

3.2. Modeling sentence embeddings into complex networks

This step corresponds to the reverse process illustrated in Fig. 1. In other words, a network representing the relationship between ideas is created from the text. The construction of networks from vector structures has been explored in recent works. In [42], the authors present such a transformation as a framework in complex systems analysis. The transformation of vector structures into networks has also been used in the context of text analysis [41,43]. The creation of a complex network from Word2Vec was proposed using a twofold approach. The d-proximity technique links all nodes whose distance from the reference node is lower than d. The second technique is the k-NN approach, which links all k nearest nodes to the reference node. In the same line, [41] created a network based on word embeddings. However, the authors aimed at creating a network that takes into account the sense of words to solve ambiguities. Each occurrence of an ambiguous words was modeled as a node in the network. Nodes were represented by a vector combining the embeddings of the words in the context. Two occurrences of an ambiguous were then connected whenever the respective embeddings were similar. In other words, two ambiguous words were linked if they appeared in similar contexts.

In the current study, sentences were connected according to the k-NN technique, as suggested by other works [43]. Each sentence is represented as a vector according to Eq. (2). The value of k in the main experiments were chosen to allow that each network is composed of a single connected component. In particular, the lowest k allowing the creation of a connected network was used for each book.

3.3. Community detection

The next step in the proposed framework concerns the detection of semantic fields, i.e. the communities in the network of sentences. A recurrent phenomena in several complex networks is the existence of communities, i.e. groups of strongly connected nodes. Similarly to other network measurements, the detection of communities gives important information regarding the organization of networks. Communities are present in different networks including in biological, social and information networks [44].

A well-known measure to quantify the quality of partitions in complex networks is the modularity [45,46]. This measure compares the obtained partition with a null model, i.e. a network with similar properties but with no community structure. Several algorithms have been proposed to address the community detection problem via optimization of the modularity. In the main experiments we used the Louvain method [47] to identify communities. The main advantage of this method is its computational efficiency, which has allowed its use in several contexts [43,48]. Another advantage associated to this algorithm is that no additional parameters are required to optimize the modularity. In additional experiments, we also probed the effect of other community detection methods on the performance of the framework in the considered classification tasks. In addition to the Louvain method, we also used the following three methods: (i) fastgreedy, (ii) eigenvector, and (iii) walktrap. An introduction to these methods can be found in [44,49].

In the proposed network representation, communities represent groups of interconnected sentences about a given topic. Because the k-NN construction allows nodes to be connected to other close nodes and, considering the Word2Vec an efficient semantic representation, the linking strategy allows the creation of dense clusters of semantically related sentences. This idea of semantic clusters has also been explored via community detection in similar works [41,43,50]. For example, using networks built at the word level, the groups detected in [43] were found to represent large cities, professions and others topics. In [41], the obtained groups were found to represent words conveying the same sense.

In order to illustrate the process of obtaining semantic communities, we performed an analysis of the obtained communities in the book “Alice’s Adventures in Wonderland”, by Lewis Carroll (see Fig. 3). We summarize below the main topics approaches in some of the communities obtained by the Louvain algorithm:

1. *Community A*: this community includes sentences mentioning animals (e.g. “pet”, “cat”, “mouse” and “dog”). This community also includes dialogs between Alice and animals. “Cat” is the main character in this community.
2. *Community B*: this community includes words sentiment words expressed via speeches. Some of the words in this community are “passionate”, “melancholy”, “angrily”, “shouted” and “screamed”.
3. *Community C*: this community includes several adverbs related to Alice’s actions.
4. *Community D*: this community includes words related to sentiments such as anger, tranquility and peacefulness.
5. *Community E*: this community is most related to the word “soup”.
6. *Community F*: this community is related to geographical locations, including countries and cities (Australia, Rome and New Zealand). Interestingly, this community also included the word “Cricket”, a prominent sport in Australia.
7. *Community G*: this community included mostly sentences referring to “Dormouse”, one of the main characters in the plot.

While most of the obtained communities are informative, a few communities were found to be more dispersed, approaching more than one topic. This might occur given the limitations of the embeddings model, since some words might not be available in the considered model. Despite these limitations, we show that the flow of information (from sentence to sentence) in the obtained semantic communities can be used to characterize texts.

3.4. Markov chains

In order to capture how authors move from community to community (semantic field) as their story unfolds, we create a representation of community transitions. The idea of studying language via Markov process is not recent. One of the first uses of this model is the study of letters sequences [51]. Since then, Markov chains are used as a statistical tool in several natural language processing problems, including language modeling, machine translation and speech recognition [52].

Here we represented the transitions between semantic fields (network communities) as a first order Markov chain. In this representation, each community becomes a state. Note that this approach of representing communities as a single unit has also been used in other contexts [15]. The probabilities of transition are considered according to the frequency of transitions observed in adjacent sentences. As we shall show, using this model, it is possible to detect patterns of how authors change topics in their stories. As a proof of principle, these patterns are used to characterize texts in distinct classification tasks.

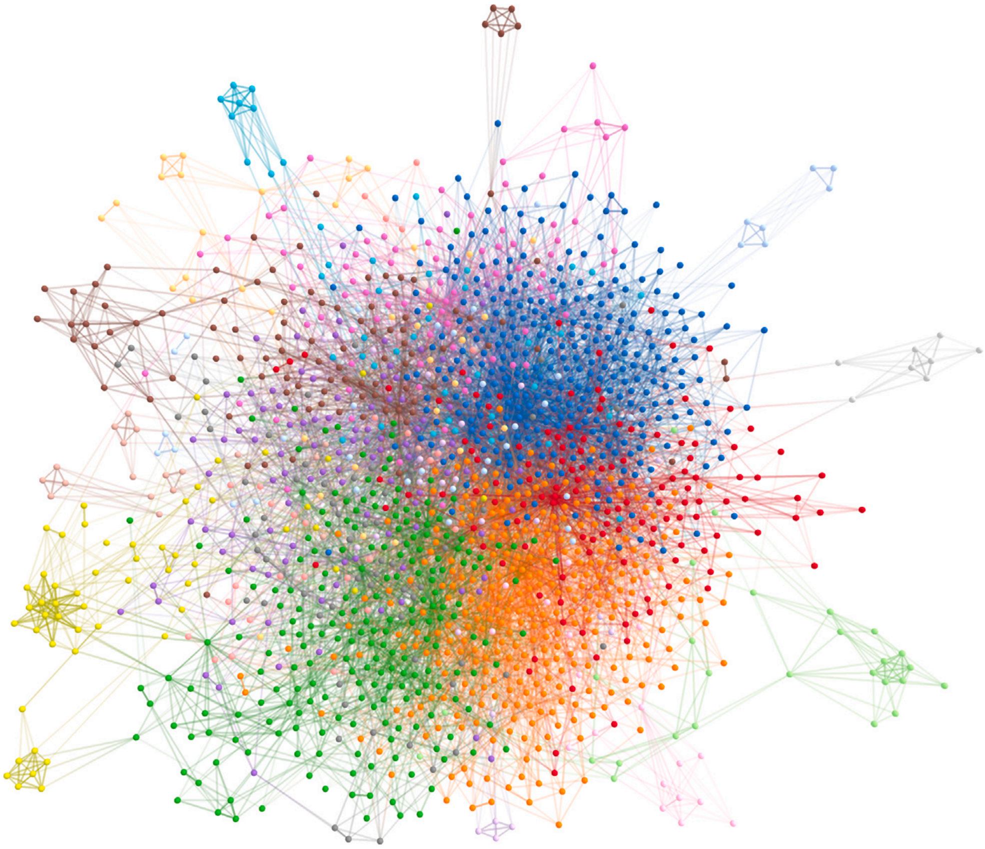


Fig. 3. Example of sentence network obtained from the book "Alice's Adventures in Wonderland", by Lewis Carroll. Colors represent community labels obtained with the Louvain method. The visualization was obtained with the method described in [15].

The process of creating a Markov chain from a network divided into communities is shown in Fig. 4. In the previous phase, communities are identified to represent distinct semantic field of the story (see left graph in Fig. 4). Because each sentence belongs to just one community, the text can be regarded as discrete time series, where each element corresponds to the membership (community label) of each sentence. Using this sequence of community labels, it is possible to create a Markov chain representing all transitions between communities (see graph on the top left of Fig. 4). Transitions weights are proportional to the frequency in which they occur and normalized so as to represent a probability. This representation is akin to a Markov chain used in other works addressing the language modeling problem [53]. The main difference here is that we are not interested in the use of particular words, but in semantic fields [54]. Once the Markov chain is obtained, we characterize this structure using *network motifs*. Note that the obtained Markov chain is equivalent to a weighted and directed complex network. Thus, traditional network tools can be used to identify network motifs [55,56].

3.5. Motifs

Network motifs are used to analyze a wide range of complex systems, including in biological, social and information networks [57]. Motifs can be defined as small subgraphs (see Fig. 5) occurring in real systems in a significant way. To quantify the significance, in general, one assumes an equivalent random network as null model. In text analysis, motifs have been used to analyze word adjacency networks in applications focusing on the syntax and style of texts [58]. More recently, an approach based on labeled motifs showed that authors tend to use words in combination with particular motifs [59]. Examples of considered motifs are represented in Fig. 5. Mathematically, the frequency of the motif with nodes "i", "j" and "k" in Fig. 5 can be computed as:

$$f_m = \sum_i \sum_j \sum_k a_{ki} a_{kj} a_{ji}, \tag{3}$$

where a_{ij} is an element of the adjacency matrix (i.e. $a_{ij} = 1$ if there is an edge from i to j and $a_{ij} = 0$, otherwise). A similar equation can be used to compute the frequency of all possible motifs comprising three nodes. In very large networks, efficient strategies for motif discovery have been proposed [60,61].

While the structure of the Markov Chains could be analyzed using traditional network measurements, we decided not to use these measurements owing to the limited size of these structures. As suggested in related works, a characterization

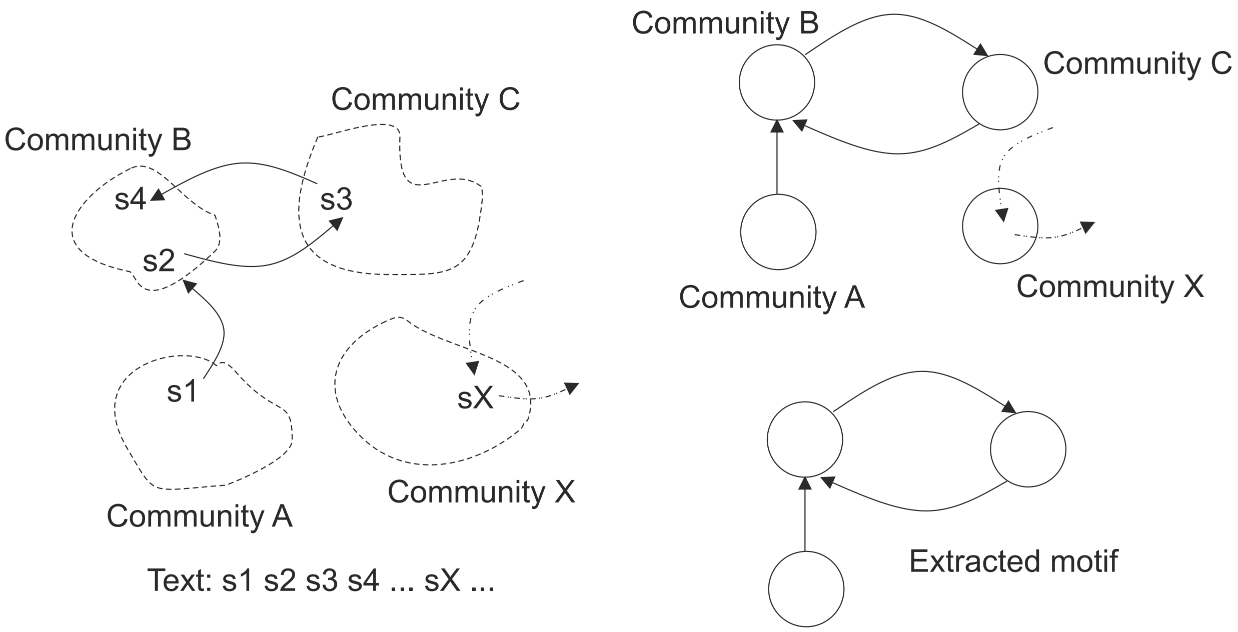


Fig. 4. Example of extraction of motifs from the network. As the text unfolds according to a given order of sentences ($s1, s2, s3, s4 \ldots sX \ldots$) a sequence of communities is generated (Community A, Community B, Community C, Community B). This sequence is used to create a Markov chain. Finally, the Markov chain is characterized by counting different patterns (motifs) of community transitions.

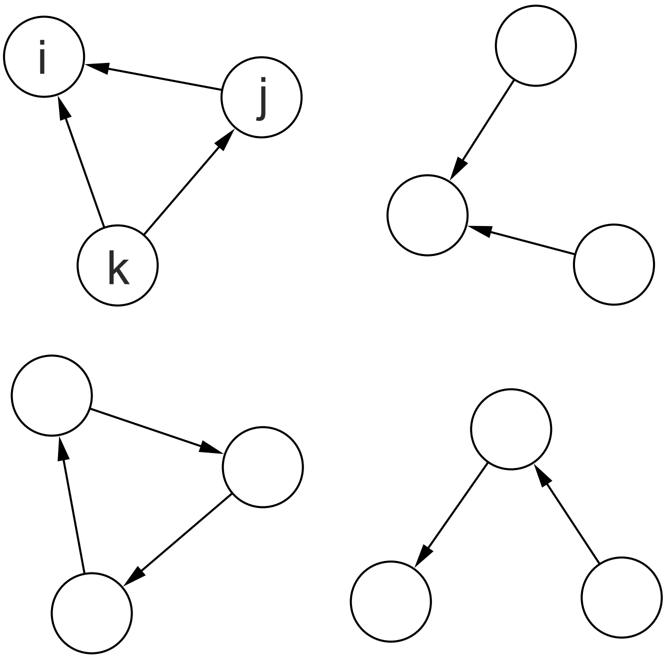


Fig. 5. Example of representative motifs comprising three nodes. The frequency of occurrence of the motif in the left upper corner can be computed using Eq. (3).

based on network metrics in small networks might not be informative [20,62,63]. As we shall show in the results, this is a simple, yet useful approach to classify small Markov Chains. In the results section, we only show the results obtained for three-node connected motifs. In preliminary experiments, no significant gain was observed when considering network motifs comprising four nodes.

The following approaches were considered to extract motifs from Markov networks. We discriminated strategies according to the use of weights to count motifs (unweighted vs weighted). If a thresholding is applied before extracting motifs, the strategy is referred to as "simplified" (see strategies 2–4 below).

1. *Unweighted strategy*: no thresholding is applied. All weights are disregarded. Every time a motif is detected, its frequency is increased by one.

Table 1
Accuracy rate and p-value obtained for the classification subtasks. Only the best results are shown among all considered classifiers. We considered the *unweighted* version of the Markov networks to extract motifs.

Subtask	Acc.	p-value
Children × investigative	50.8%	5.56×10^{-1}
Children × philosophy	71.6%	1.30×10^{-3}
Investigative × philosophy	70.8%	3.30×10^{-3}
Children × investigative × philosophy	43.8%	3.50×10^{-2}

2. *Simplified unweighted strategy*: this approach is the same as the *unweighted* strategy. However, before counting motifs, the weakest edges are removed according to a given threshold.
3. *Simplified weighted strategy*: before counting motifs, the weakest edges are removed according to a given threshold. Then motifs are identified disregarding edges weights (e.g. using Eq. (3)). Edge weights are then considered to update the frequency associated to that motif. Every time a given motif is found, the respective "frequency" of that motif is increased by the sum of the weights of its edges.
4. *Simplified weighted strategy with local thresholding*: this technique is similar to the *simplified weighted strategy*. However, here a different approach is used to threshold the networks. We consider here a local threshold, which is established for each edge. A local thresholding strategy is important for some network applications because a simple global threshold value might overlook important network structures, such as network communities [64–66].
The local thresholding strategy proposed in [66] evaluates the relevance of an edge by computing a p-value determined in terms of a null model. The null model computes the likelihood of a node v having an edge with a specific weight by taking into account the other edges connected to v. In practice, the relevance α_{ij} of an edge e_{ij} connecting nodes i and j is computed as:

$$\alpha_{ij} = 1 - (k_i - 1) \int_0^{\pi_{ij}} (1-x)^{k_i - 2} dx, \tag{4}$$

$$\pi_{ij} = w_{ij} \left(\sum_{ik \in E} w_{ik} \right)^{-1}, \tag{5}$$

where w_{ij} is the weight of e_{ij} and $k_i = \sum_j a_{ij}$. All edges with $\alpha_{ij} < \mathcal{A}$ are removed from the network, where $\mathcal{A}$ is the adopted local threshold.

3.6. Classification

The extracted motifs from the Markov Chains are used as input (features) to the classification systems. The following methods were used in the experiments: Decision Tree (CART), kNN, SVM (linear) and Naive Bayes [67]. The evaluation was performed using a 10-fold cross-validation approach. As suggested in related works, all classifiers were trained with their default configuration of parameters [68].

4. Results and discussion

Here we probed whether the dynamics of changes in semantic groups in books can be used to characterize stories. The proposed methodology was applied in two distinct classification tasks. In the first task, we aimed at distinguishing three different thematic classes: (i) children books; (ii) investigative; and (iii) philosophy books. The second aimed at discriminating books according to their publication dates. All books (and their respective classes) were obtained from the Gutenberg repository. The list of books and respective authors are listed in the Supplementary Information. The corpora size is compatible with other works in the literature [18,63,69,70].

In the first experiment, we evaluated if patterns of semantic changes are able to distinguish between children, philosophy or investigative books. We considered problems with two or three classes. The obtained results are shown in Table 1. In this case, weights were disregarded after the construction of the Markov networks (*unweighted* version). Considering subtasks encompassing only two classes, only the distinction between children and investigative texts were not significant, with a low accuracy rate. The distinction philosophy books and the other two classes, however, yielded a much better discrimination. These results were found to be significant. When all three classes are discriminated, a low accuracy rate was found (43.8%), even though this still represents a significant result. The low accuracy rate found using the proposed approach is a consequence of a regular behavior found in the Markov chains. In other words, in most of the books, all communities were found to be connected to each other, hampering thus the discriminability of different types of books.

Table 2
Accuracy rate and p-value obtained for the classification subtasks. Only the best results are shown among all considered classifiers and thresholds. We considered the *simplified unweighted* version of the Markov networks to extract motifs.

Subtask	Acc.	Threshold	p-value
Children × investigative	65.8%	0.060	1.64×10^{-2}
Children × philosophy	81.0%	0.190	1.19×10^{-5}
Investigative × philosophy	91.6%	0.075	2.23×10^{-10}
Children × investigative × philosophy	62.2%	0.075	2.00×10^{-7}

Table 3
Accuracy rate and p-value obtained for the classification subtasks. Only the best results are shown among all considered classifiers and thresholds. We considered the *simplified weighted* version of the Markov networks to extract motifs.

Subtask	Acc.	Threshold	p-value
Children × investigative	70.8%	0.075	3.30×10^{-3}
Children × philosophy	89.0%	0.145	1.62×10^{-8}
Investigative × philosophy	92.5%	0.120	2.23×10^{-10}
Children × investigative × philosophy	62.7%	0.075	2.00×10^{-7}

Table 4
Accuracy rate obtained for the classification subtasks, considering distinct community detection methods: Louvain, walktrap, eigenvector and fastgreedy [44]. Only the best results are shown among all considered classifiers, thresholds and community detection methods. We considered the *simplified weighted* version of the Markov networks to extract motifs. The following parameters were used in the classifiers: SVM (linear kernel and penalty parameter of the error term = 1.0), CART (criterion to measure the quality of a split = gini, minimum number of samples required to split an internal node = 2, minimum number of samples required to be at a leaf node = 1), Naive Bayes (GaussianNB) and kNN ($k = 1$, Euclidean distance).

Subtask	Acc.	Threshold	Method	Classifier
children × investigative	76.7%	0.045	Eigenvector	SVM
children × philosophy	90.8%	0.07	Walktrap	SVM
investigative × philosophy	97.5%	0.185	FastGreedy	CART
children × investigative × philosophy	70.0%	0.17	FastGreedy	kNN

Given the low accuracy rates obtained with the *unweighted* strategy, we analyzed if the *simplified* unweighted version was able to provide a better characterization. In this case, the weakest edges were removed before the extraction of motifs. We considered the thresholding ranging between 0.01 and 0.20. The main idea here is to remove less important links between communities. The obtained results are shown in Table 2. All obtained results turned out to be significant. All previous accuracy rates were improved. Interestingly, a high discrimination rate (91.6%) was obtained when discriminating investigative and philosophy books. These results suggest that the threshold is an important pre-processing step here, given that it can boost the performance of the classification by a large margin.

When combining thresholding and edges weights in the *simplified weighted* version, the results obtained in Table 3 were further improved. The highest gain in performance was observed when discriminating children from philosophy books: the performance improved from 81.0% to 89.0%. Only a minor improvement was observed when all three classes were discriminated. Overall, this results suggest that both thresholding and the use of edges weights might be useful to characterize Markov networks. Most importantly, all three methods showed that, in fact, there is a correlation between the thematic approached and the way in which authors approaches semantic groups in texts.

While the main focus of this manuscript is not to provide the best combination to optimize the performance of a classification task, it is still interesting to probe how the classification based on the concept of semantic flow can benefit from different partitions (semantic clusters) extracted from different community detection methods. The best results obtained by comparing 4 distinct methods are summarized in Table 4. Interestingly, note that an impressive 97.5% accuracy rate was observed when discriminating investigative and philosophy books. In this case, a feature relevance analysis revealed that two particular motifs are responsible for most of the discriminative power (see analysis in Fig. 6). Additional results considering the strategy based on *unweighted* motifs strategy are shown in Table S1 of the Supplementary Information.

The discriminative power of the obtained networks was also investigated using a local strategy to threshold the network. In other words, the relevance of an edge in the *simplified weighted strategy with local thresholding* depends on the weights of its neighboring edges (see Section 3.5) [66]. We show in Table 5 the results obtained with this technique when adopting as local threshold the value $\mathcal{A} = 0.95$. We found that, for this particular technique, the results are not improved, even when other values for $\mathcal{A}$ are considered (additional results are shown in the Supplementary Information). For this reason, we did not consider the *simplified weighted strategy with local thresholding* in the next results.

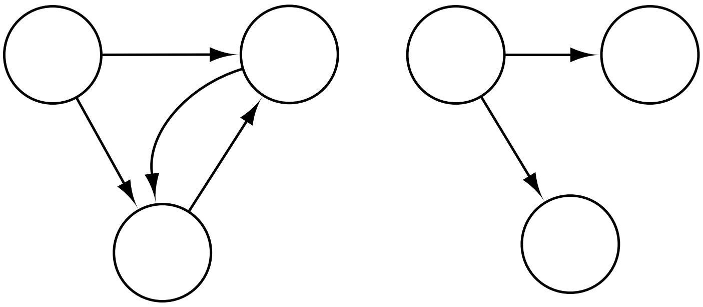


Fig. 6. The above motifs are responsible for most of the discriminative power considering the task "investigative x philosophy" in the configuration of methods and parameters used in Table 4. In order to compute relevance, we used the Gini index [71] (Γ) in the CART classification method. The obtained relevance values for the motifs on the left and right sides of the figure are, respectively, $\Gamma = 0.809$ and $\Gamma = 0.147$. While the relevance of features (i.e. motifs) may differ in other classification tasks, the above motifs were found to be among the most discriminative ones in the classification tasks studied in this paper (result not shown).

Table 5
Accuracy rate and *p*-value obtained for the classification subtasks. We considered the *simplified weighted strategy with local thresholding* to extract motifs. All edges with $\alpha_{ij} < \mathcal{A}$ were removed (see Eq. (4)). The obtained performance is no better than the one obtained with the global thresholding approach..

Subtask	Acc.	*p*-value
Children × investigative	63.3%	3.25×10^{-2}
Children × philosophy	66.3%	1.60×10^{-2}
Investigative × philosophy	71.3%	3.30×10^{-3}
Children × investigative × philosophy	48.0%	5.94×10^{-3}

Table 6
Performance of the proposed method using the *simplified unweighted* motif characterization of Markov networks. For each subtask, only the best threshold obtained for the best classifier is shown.

Subtask	Acc.	Threshold	*p*-value
1700–1799 × 1800–1899	70.0%	0.195	1.34×10^{-3}
1700–1799 × 1900 or later	75.0%	0.060	6.73×10^{-5}
1800–1899 × 1900 or later	70.0%	0.160	1.34×10^{-3}
1700–1850 × 1851 or later	66.0%	0.010	6.74×10^{-3}
1700–1799 × 1800–1899 × 1900 or later	55.0%	0.025	1.22×10^{-5}

In order to investigate the dependency of the classification results on the word embeddings model, we also considered embeddings obtained from the BERT model [39]. While there is only a minor improvement in particular cases, the results obtained with the Word2vec model (see Table 4) provides most of the best results. The results obtained with the BERT model are summarized in Tables S2 and S3 of the Supplementary Information.

We also investigated if the patterns of semantic flow varies with the publication date. For this reason, we selected a dataset with books in different periods. The following classes were considered, according to the range of publication dates:

1. Books published between 1700 and 1799.
2. Books published between 1800 and 1899.
3. Books published after 1900.
4. Books published between 1700 and 1850.
5. Books published after 1851.

The results obtained in the classification for different subtasks is shown in Table 6. We only show here the results obtained for the simplified unweighted characterization because it yielded the best results. Overall, all classification results are significant, confirming thus that there are statistically significant differences of semantic flow patterns for books published in different epochs. However, the results obtained here are worse than the ones obtained in the dataset with books about different themes (see Table 3). Therefore, patterns of semantic flow seems to be less affected by the year of publication, while being more sensitive to the subject/topic approached by the text.

5. Conclusion

In this paper we investigate whether patterns of semantic flow arises for different classes of texts. To represent the relationship between ideas in texts, we used a sentence network representation, where sentences (nodes) are connected based on their semantic similarity. Semantic clusters were identified via community detection and high-level representation of each book was created based on the transition between communities as the story unfolds. Finally, motifs were extracted to characterize the patterns of transition between semantic groups (communities). When applied in two distinct tasks, interesting results were found. In the task aiming at classifying books according to the approached themes, we found an high accuracy rate (92.5%) when discriminating investigative and philosophy books. A significant performance in the classification was also obtained when discriminating books published in distinct epochs. However, the discriminability for this task was not as high as the ones obtained when discriminating investigative, philosophy and children books.

Given the complexity of the components in the proposed framework, we decided not to optimize each step of the process. Even without a rigorous optimization process, we were able to identify semantic flow patterns that were able to discriminate distinct classes of texts. As future works, we intend to perform a systematic analysis on how to optimize the process. For example, during the construction of the networks, different approaches could be used to create embeddings and link similar sentences [72]. In a similar fashion, different strategies to identify communities could also be used in the analysis. Finally, we could also investigate additional approaches to characterize the obtained Markov networks.

The proposed framework identified clusters of ideas being conveyed in texts. We basically measured, for each story, how authors move from one semantic cluster to another while the story is being told. This gives the sense of "semantic flow" measured in terms of network motifs. Our results suggest, therefore, that different classes of stories have distinct semantic flow patterns. For example, in the classification of children and philosophical books, one should expect that the dynamics of changing topics in children books should be much less complex than the semantic flow observed in books about philosophy. Such a difference could be related to the cognitive efforts required to the reader to understand different patterns of semantic flow. Concerning the classification based on publication dates, the high discriminability could be related to the fact that each century is characterized by a different style. These are hypothesis that should be evaluated in future works by using potential available datasets.

Our results suggest that semantic flow motifs could play an important role in other NLP tasks. For example, in the authorship recognition task, patterns extracted from a semantic flow analysis could be combined with other techniques to improve the characterization of authors [73,74]. In fact, the use of motifs in the microscopic level has already provided a good characterization of authors [59]. A similar idea could also be applied to the analysis of other stylometric tasks. In addition, we suggest that the semantics of the texts could be combined with the concept of semantic flow by using "labeled motifs", as proposed in our previous work [59]. Since semantic networks have been studied in cognitive sciences, we believe that the adopted network representation could be adapted and used – as an auxiliary tool – to study complex brain and cognitive processes that could assist the diagnosis of cognitive disorders via text analysis [73,75].

CRediT authorship contribution statement

Edilson A. Corrêa Jr.: Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Vanessa Q. Marinho:** Software, Validation, Investigation. **Diego R. Amancio:** Conceptualization, Methodology, Validation, Formal analysis, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

E.A.C. Jr. and D.R.A. acknowledge financial support from Google (Google Research Awards in Latin America grant). V.Q.M. acknowledges financial support from São Paulo Research Foundation (FAPESP) (Grant no. 15/05676-8). D.R.A. also thanks FAPESP (Grant no. 16/19069-9) and CNPq-Brazil (Grant no. 304026/2018-2) for support.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.physa.2020.124895.

References

[1] W. Jin, R.K. Srihari, Graph-based text representation and knowledge discovery, in: Proceedings of the 2007 ACM Symposium on Applied Computing, ACM, 2007, pp. 807–811.
[2] R.F. Cancho, R.V. Solé, R. Köhler, Patterns in syntactic dependency networks, Phys. Rev. E 69 (5) (2004) 051915.
[3] M.A. Montemurro, D.H. Zanette, Keywords and co-occurrence patterns in the voynich manuscript: An information-theoretic analysis, PLoS One 8 (6) (2013) e66344.
[4] K. Ban, M. Perc, Z. Levnajić, Robust clustering of languages across wikipedia growth, R. Soc. Open Sci. 4 (10) (2017) 171217.

[5] B. Tadić, M. Andjelković, B.M. Boshkoska, Z. Levnajić, Algebraic topology of multi-brain connectivity networks reveals dissimilarity in functional patterns during spoken communications, PLoS One 11 (11) (2016) e0166787.
[6] D.R. Amancio, F.N. Silva, L.F. Costa, Concentric network symmetry grasps authors' styles in word adjacency networks, Europhys. Lett. 110 (6) (2015) 68001.
[7] M. Stella, A. Zaytseva, Forma mentis networks map how nursing and engineering students enhance their mindsets about innovation and health during professional growth, PeerJ Comput. Sci. 6 (2020) e255.
[8] N. Castro, M. Stella, The multiplex structure of the mental lexicon influences picture naming in people with aphasia, J. Complex Netw. 7 (6) (2019) 913–931.
[9] M. Stella, Modelling early word acquisition through multiplex lexical networks and machine learning, Big Data Cogn. Comput. 3 (1) (2019) 10.
[10] M. Stella, S. De Nigris, A. Aloric, C.S. Siew, Forma mentis networks quantify crucial differences in STEM perception between students and experts, PLoS One 14 (10) (2019).
[11] 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.
[12] D.R. Amancio, O.N. Oliveira Jr., L.F. Costa, Unveiling the relationship between complex networks metrics and word senses, Europhys. Lett. 98 (1) (2012) 18002.
[13] D.R. Amancio, O.N. Oliveira Jr, L.F. Costa, Identification of literary movements using complex networks to represent texts, New J. Phys. 14 (4) (2012) 043029.
[14] D.R. Amancio, S.M. Aluisio, O.N. Oliveira Jr, L.F. Costa, Complex networks analysis of language complexity, Europhys. Lett. 100 (5) (2012) 58002.
[15] 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. Informetr. 10 (2) (2016) 487–502.
[16] R.F. Cancho, R.V. Solé, The small world of human language, Proc. R. Soc. Lond. Ser. B: Biol. Sci. 268 (1482) (2001) 2261–2265.
[17] H. Liu, J. Cong, Language clustering with word co-occurrence networks based on parallel texts, Chin. Sci. Bull. 58 (10) (2013) 1139–1144.
[18] H.F. Arruda, V.Q. Marinho, L.F. Costa, D.R. Amancio, Paragraph-based representation of texts: a complex networks approach, Inf. Process. Manage. 56 (2019) 479–494.
[19] H.F. Arruda, L.F. Costa, D.R. Amancio, Using complex networks for text classification: Discriminating informative and imaginative documents, Europhys. Lett. 113 (2) (2016) 28007.
[20] D.R. Amancio, Probing the topological properties of complex networks modeling short written texts, PLoS One 10 (2) (2015) e0118394.
[21] H.F. de Arruda, F.N. Silva, C.H. Comin, D.R. Amancio, L.F. Costa, Connecting network science and information theory, Physica A 515 (2019) 641–648.
[22] W. Ebeling, A. Neiman, Long-range correlations between letters and sentences in texts, Physica A 215 (3) (1995) 233–241.
[23] A. Schenkel, J. Zhang, Y.-C. Zhang, Long range correlation in human writings, Fractals 1 (01) (1993) 47–57.
[24] E. Alvarez-Lacalle, B. Dorow, J.-P. Eckmann, E. Moses, Hierarchical structures induce long-range dynamical correlations in written texts, Proc. Natl. Acad. Sci. 103 (21) (2006) 7956–7961.
[25] M. Amit, Y. Shmerler, E. Eisenberg, M. Abraham, N. Shnerb, Language and codification dependence of long-range correlations in texts, Fractals 2 (01) (1994) 7–13.
[26] H.F. Arruda, F.N. Silva, V.Q. Marinho, D.R. Amancio, L.F. Costa, Representation of texts as complex networks: a mesoscopic approach, J. Complex Netw. 6 (1) (2018) 125–144.
[27] H.F. Arruda, F.N. Silva, L.F. Costa, D.R. Amancio, Knowledge acquisition: A complex networks approach, Inform. Sci. 421 (2017) 154–166.
[28] J.V. Tohalino, D.R. Amancio, Extractive multi-document summarization using multilayer networks, Physica A 503 (2018) 526–539.
[29] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (Feb) (2003) 1137–1155.
[30] 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.
[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.
[32] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.
[33] 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.
[34] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 746–751.
[35] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[36] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 379–389.
[37] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[38] 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.
[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv:1810.04805.
[40] J. Camacho-Collados, M.T. Pilehvar, From word to sense embeddings: A survey on vector representations of meaning, J. Artif. Int. Res. (ISSN: 1076-9757) 63 (1) (2018) 743–788.
[41] E.A. Corrêa Jr, D.R. Amancio, Word sense induction using word embeddings and community detection in complex networks, Physica A 523 (2019) 180–190.
[42] C.H. Comin, T. Peron, F.N. Silva, D.R. Amancio, F.A. Rodrigues, L.d.F. Costa, Complex systems: features, similarity and connectivity, Phys. Rep. (2020).
[43] 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.
[44] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[45] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[46] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[47] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Statist. Mech.: Theory Exp. 2008 (10) (2008) P10008.
[48] E. Corrêa Jr., A.A. Lopes, D.R. Amancio, Word sense disambiguation: A complex network approach, Inform. Sci. 442–443 (2018) 103–113.
[49] M. Newman, Networks: An Introduction, Oxford University Press, Inc., New York, NY, USA, 2010.

[50] L. Antiqueira, O.N. Oliveira Jr, L. da Fontoura Costa, M.d.G.V. Nunes, A complex network approach to text summarization, Inform. Sci. 179 (5) (2009) 584–599.

[51] A.A. Markov, An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains, Proc. Bibliogr. Acad. Sci. 7 (6) (1913) 153–162.

[52] C.D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, USA, ISBN: 0-262-13360-1, 1999.

[53] J.M. Ponte, W. Croft, A language modeling approach to information retrieval, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 275–281.

[54] F. Li, T. Dong, Text categorization based on semantic cluster-hidden markov models, in: International Conference in Swarm Intelligence, Springer, 2013, pp. 200–207.

[55] S. Wernicke, F. Rasche, FANMOD: a tool for fast network motif detection, Bioinformatics 22 (9) (2006) 1152–1153.

[56] S. Omidi, F. Schreiber, A. Masoudi-Nejad, MODA: an efficient algorithm for network motif discovery in biological networks, Genes Genet. Syst. 84 (5) (2009) 385–395.

[57] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (5594) (2002) 824–827.

[58] D.R. Amancio, E.G. Altmann, D. Rybski, O.N. Oliveira Jr, L.F. Costa, Probing the statistical properties of unknown texts: application to the voynich manuscript, PLoS One 8 (7) (2013) e67310.

[59] V.Q. Marinho, G. Hirst, D.R. Amancio, Labelled network subgraphs reveal stylistic subtleties in written texts, J. Complex Netw. 6 (4) (2018) 620–638.

[60] Y. Kavurucu, A comparative study on network motif discovery algorithms, Int. J. Data Min. Bioinform. 11 (2) (2015) 180—204.

[61] S. Wernicke, Efficient detection of network motifs, IEEE/ACM Trans. Comput. Biol. Bioinform. 3 (4) (2006) 347–359.

[62] B.C. Van Wijk, C.J. Stam, A. Daffertshofer, Comparing brain networks of different size and connectivity density using graph theory, PLoS One 5 (10) (2010) e13701.

[63] D.R. Amancio, Comparing the topological properties of real and artificially generated scientific manuscripts, Scientometrics 105 (3) (2015) 1763–1779.

[64] M. Tumminello, S. Micciche, F. Lillo, J. Piilo, R.N. Mantegna, Statistically validated networks in bipartite complex systems, PLoS One 6 (3) (2011).

[65] F. Radicchi, J.J. Ramasco, S. Fortunato, Information filtering in complex weighted networks, Phys. Rev. E 83 (4) (2011) 046101.

[66] M.Á. Serrano, M. Boguná, A. Vespignani, Extracting the multiscale backbone of complex weighted networks, Proc. Natl. Acad. Sci. 106 (16) (2009) 6483–6488.

[67] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[68] M.Z. Rodriguez, C.H. Comin, D. Casanova, O.M. Bruno, D.R. Amancio, L.d.F. Costa, F.A. Rodrigues, Clustering algorithms: A comparative approach, PLoS One 14 (1) (2019).

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

[70] S. Segarra, M. Eisen, G. Egan, A. Ribeiro, Attributing the authorship of the henry VI plays by word adjacency, Shakespear. Quart. 67 (2) (2016) 232–256.

[71] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Syst. Appl. (ISSN: 0957-4174) 33 (1) (2007) 1–5.

[72] R. Kiros, Y. Zhu, R.R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, in: Advances in Neural Information Processing Systems, 2015, pp. 3294–3302.

[73] A. Baronchelli, R. Ferrer-i Cancho, R. Pastor-Satorras, N. Chater, M.H. Christiansen, Networks in cognitive science, Trends Cogn. Sci. 17 (7) (2013) 348–360.

[74] E.A. Corrêa Jr, F.N. Silva, L.F. Costa, D.R. Amancio, Patterns of authors contribution in scientific manuscripts, J. Informetr. 11 (2) (2017) 498–510.

[75] C.T. Kello, G.D. Brown, R. Ferrer-i Cancho, J.G. Holden, K. Linkenkaer-Hansen, T. Rhodes, G.C. Van Orden, Scaling laws in cognitive sciences, Trends Cogn. Sci. 14 (5) (2010) 223–232.

CHAPTER 5

CONCLUSION

The work of this thesis occurred at the same time that the success of deep learning was spreading through several tasks in different fields. In the area of NLP, disregarding the period when neural networks were used as just another alternative for machine learning methods, the heavy use of deep learning started with methods to generate word embeddings such as Word2Vec (MIKOLOV *et al.*, 2013a; MIKOLOV *et al.*, 2013b), GloVe (PENNINGTON; SOCHER; MANNING, 2014), Wang2Vec (LING *et al.*, 2015) and fastText (JOULIN *et al.*, 2017). Soon after, some methods capable of generating embeddings for larger structures such as sentences, paragraphs and even complete texts emerged, some of these techniques used the compositionality of word vectors (combination of embeddings, technique used in two of the works of this thesis), others through the use of sequential models, like the seq2seq model for machine translation (SUTSKEVER; VINYALS; LE, 2014) or for general purpose as in the case of Skip-thought (KIROS *et al.*, 2015) and fastText (JOULIN *et al.*, 2016). Along with the widespread use of sequential models, attention mechanisms (BAHDANAU; CHO; BENGIO, 2015) were created, which allowed sequential models to give 'attention' according to the specific needs of the problem, thus generating higher quality embeddings. Its success led to the creation of the Transformer (VASWANI *et al.*, 2017), a network architecture which has as its basic principle the application of attention throughout its entire architecture. With this new architecture, a new generation of methods capable of generating general purpose embeddings (for word, sentence and larger structures) and also to perform end-to-end tasks like BERT (DEVLIN *et al.*, 2018) and GPT2 (RADFORD *et al.*, 2019). Pre-trained language models that now are widely adopted in wide range of NLP tasks and applications.

At the beginning of our work, we started attacking context modeling only with the use of machine learning models that were based on complex networks, but given the trends and research directions in NLP, we realized that the combination of techniques from deep learning it would not only bring a significant gain to our work but would also explore an intersection between areas yet to be consolidated. Thus, of the various models and works on deep learning available, we use both pre-trained word embeddings and more complex models such as BERT. Always following the hypothesis that a good modeling and understanding of the context would lead to better results in NLP tasks. A hypothesis that proved to be fruitful in our work, the use of word embeddings allowed a good modeling of the local context and the use of a complex network structure allowed the representation of different levels of context and also allowed the application of community detection methods, operating as an alternative to traditional unsupervised methods (clustering), whether for its use in disambiguation or for the classification of texts.

In addition to the specific "further works" of each article, we consider that some more general studies should be considered in the future, like the devise of new **network formation** approaches. The way in which a network is set up directly influences how measures and methods behave in a complex network, which leads us to question whether it would be possible to create processes capable of considering important measures such as modularity, which could benefit the application of community detection methods in the structure; **similarity learning**, even in the context of network modeling, the use of established distances in the literature is common, but learning similarity that adheres to the problem in question could significantly improve representation in networks; **context expansion**, the inclusion of random or even task related information can help the network structure to highlight context information, as demonstrated in the Word2Vec method, the inclusion of random information, if done correctly, can lead to greater power of discrimination of machine learning models, thus, it is possible to consider that in a network of ambiguous words (represented by context) having an addition of random content (contexts unrelated to the ambiguous word) could better represent community structures, thus bringing a improvement of the community detection method; **new tasks**, in general, our work not only presents some techniques, but a framework that can be explored, expanded and modified to measure, thus allowing its application in several unsupervised NLP tasks that use the context implicitly or explicitly.

BIBLIOGRAPHY

ADHIKARI, B.; ZHANG, Y.; RAMAKRISHNAN, N.; PRAKASH, B. A. Distributed representation of subgraphs. **arXiv preprint arXiv:1702.06921**, 2017. Citation on page 91.

AGIRRE, E.; EDMONDS, P. **Word sense disambiguation: Algorithms and applications**. [S.l.]: Springer Science & Business Media, 2007. Citation on page 79.

AMANCIO, D. R. **Classificação de textos com redes complexas**. Phd Thesis (PhD Thesis) — Instituto de Física de São Carlos - Universidade de São Paulo, 2013. Citation on page 79.

AMANCIO, D. R.; ALTMANN, E. G.; JR, O. N. O.; COSTA, L. da F. Comparing intermittency and network measurements of words and their dependence on authorship. **New Journal of Physics**, IOP Publishing, v. 13, n. 12, p. 123024, 2011. Citation on page 79.

AMANCIO, D. R.; ANTIQUEIRA, L.; PARDO, T. A.; COSTA, L. da F.; JR, O. N. O.; NUNES, M. G. Complex networks analysis of manual and machine translations. **International Journal of Modern Physics C**, World Scientific, v. 19, n. 04, p. 583–598, 2008. Citation on page 79.

AMANCIO, D. R.; JR, O. N. O.; COSTA, L. d. F. Unveiling the relationship between complex networks metrics and word senses. **EPL (Europhysics Letters)**, IOP Publishing, v. 98, n. 1, p. 18002, 2012. Citations on pages 16, 19, and 79.

ANTIQUEIRA, L.; OLIVEIRA, O. N.; COSTA, L. da F.; NUNES, M. d. G. V. A complex network approach to text summarization. **Information Sciences**, Elsevier, v. 179, n. 5, p. 584–599, 2009. Citation on page 79.

ARRUDA, H. F. de; COSTA, L. d. F.; AMANCIO, D. R. Using complex networks for text classification: Discriminating informative and imaginative documents. **EPL (Europhysics Letters)**, IOP Publishing, v. 113, n. 2, p. 28007, 2016. Citations on pages 16 and 79.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. In: **3rd International Conference on Learning Representations, ICLR 2015**. [S.l.: s.n.], 2015. Citations on pages 16, 17, 63, and 91.

BARONCHELLI, A.; CANCHO, R. Ferrer-i; PASTOR-SATORRAS, R.; CHATER, N.; CHRISTIANSEN, M. H. Networks in cognitive science. **Trends in cognitive sciences**, Elsevier, v. 17, n. 7, p. 348–360, 2013. Citation on page 16.

BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: **ACL (1)**. [S.l.: s.n.], 2014. p. 238–247. Citation on page 91.

BAYER, J.; WIERSTRA, D.; TOGELIUS, J.; SCHMIDHUBER, J. Evolving memory cell structures for sequence learning. In: SPRINGER. **International Conference on Artificial Neural Networks**. [S.l.], 2009. p. 755–764. Citation on page 87.

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **Journal of Machine Learning Research**, v. 3, n. Feb, p. 1137–1155, 2003. Citations on pages 16, 89, and 90.

BENGIO, Y.; LAMBLIN, P.; POPOVICI, D.; LAROCHELLE, H. *et al.* Greedy layer-wise training of deep networks. **Advances in neural information processing systems**, MIT; 1998, v. 19, p. 153, 2007. Citation on page 87.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994. Citation on page 86.

BIEMANN, C. Unsupervised part-of-speech tagging employing efficient graph clustering. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: student research workshop**. [S.l.], 2006. p. 7–12. Citation on page 79.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. **Journal of statistical mechanics: theory and experiment**, IOP Publishing, v. 2008, n. 10, p. P10008, 2008. Citation on page 78.

BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks: Structure and dynamics. **Physics reports**, Elsevier, v. 424, n. 4, p. 175–308, 2006. Citation on page 75.

CHO, K.; MERRIËNBOER, B. V.; BAHDANAU, D.; BENGIO, Y. On the properties of neural machine translation: Encoder-decoder approaches. **arXiv preprint arXiv:1409.1259**, 2014. Citation on page 87.

CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014. Citation on page 91.

COLLINS, M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10**. [S.l.], 2002. p. 1–8. Citation on page 89.

COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: ACM. **Proceedings of the 25th international conference on Machine learning**. [S.l.], 2008. p. 160–167. Citations on pages 16 and 90.

COLLOBERT, R.; WESTON, J.; BOTTOU, L.; KARLEN, M.; KAVUKCUOGLU, K.; KUKSA, P. Natural language processing (almost) from scratch. **Journal of Machine Learning Research**, v. 12, n. Aug, p. 2493–2537, 2011. Citation on page 90.

COMIN, C. H.; PERON, T.; SILVA, F. N.; AMANCIO, D. R.; RODRIGUES, F. A.; COSTA, L. d. F. Complex systems: features, similarity and connectivity. **Physics Reports**, Elsevier, 2020. Citation on page 79.

CONG, J.; LIU, H. Approaching human language with complex networks. **Physics of life reviews**, Elsevier, v. 11, n. 4, p. 598–618, 2014. Citation on page 16.

CORRÊA, E. A.; MARINHO, V. Q.; SANTOS, L. B. dos; BERTAGLIA, T. F. C.; TREVISO, M. V.; BRUM, H. B. Pelesent: Cross-domain polarity classification using distant supervision. In: IEEE. **2017 Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2017. p. 49–54. Citation on page 16.

COSTA, L. d. F.; JR, O. N. O.; TRAVIESO, G.; RODRIGUES, F. A.; BOAS, P. R. V.; ANTIQUEIRA, L.; VIANA, M. P.; ROCHA, L. E. C. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. **Advances in Physics**, Taylor & Francis, v. 60, n. 3, p. 329–412, 2011. Citations on pages 16 and 75.

COSTA, L. d. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citations on pages 16, 75, 76, and 77.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of Control, Signals, and Systems (MCSS)**, Springer, v. 2, n. 4, p. 303–314, 1989. Citation on page 87.

DAI, A. M.; LE, Q. V. Semi-supervised sequence learning. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2015. p. 3079–3087. Citation on page 91.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Citation on page 63.

DING, W.; LIN, C.; ISHWAR, P. Node embedding via word embedding for network community discovery. **arXiv preprint arXiv:1611.03028**, 2016. Citation on page 91.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of Machine Learning Research**, v. 12, n. Jul, p. 2121–2159, 2011. Citation on page 84.

ESTEVA, A.; KUPREL, B.; NOVOA, R. A.; KO, J.; SWETTER, S. M.; BLAU, H. M.; THRUN, S. Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, Nature Research, v. 542, n. 7639, p. 115–118, 2017. Citation on page 81.

FAN, Y.; QIAN, Y.; XIE, F.-L.; SOONG, F. K. Tts synthesis with bidirectional lstm based recurrent neural networks. In: **Interspeech**. [S.l.: s.n.], 2014. p. 1964–1968. Citation on page 87.

FIRTH, J. R. A synopsis of linguistic theory, 1930-1955. **Studies in linguistic analysis**, Basil Blackwell, 1957. Citation on page 15.

FORTUNATO, S. Community detection in graphs. **Physics reports**, Elsevier, v. 486, n. 3, p. 75–174, 2010. Citation on page 78.

GAN, Z.; PU, Y.; HENAO, R.; LI, C.; HE, X.; CARIN, L. Unsupervised learning of sentence representations using convolutional neural networks. **arXiv preprint arXiv:1611.07897**, 2016. Citation on page 91.

GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002. Citation on page 77.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citations on pages 81, 82, 84, 86, 87, and 88.

GRAVES, A.; LIWICKI, M.; FERNÁNDEZ, S.; BERTOLAMI, R.; BUNKE, H.; SCHMIDHUBER, J. A novel connectionist system for unconstrained handwriting recognition. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 31, n. 5, p. 855–868, 2009. Citation on page 87.

GROSSBERG, S. Some networks that can learn, remember, and reproduce any number of complicated space-time. **Studies in Applied Mathematics**, Wiley Online Library, v. 49, n. 2, p. 135–166, 1970. Citation on page 82.

GROVER, A.; LESKOVEC, J. node2vec: Scalable feature learning for networks. In: ACM. **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.], 2016. p. 855–864. Citation on page 91.

HAYKIN, S. S. **Neural networks: a comprehensive foundation**. [S.l.]: Tsinghua University Press, 2001. Citations on pages 82, 84, and 85.

HEBB, D. O. **The organization of behavior: A neuropsychological approach**. New York: Wiley, 1949. Citation on page 82.

HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-r.; JAITLY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N. *et al.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. **IEEE Signal Processing Magazine**, IEEE, v. 29, n. 6, p. 82–97, 2012. Citation on page 81.

HINTON, G. E. Learning distributed representations of concepts. In: AMHERST, MA. **Proceedings of the eighth annual conference of the cognitive science society**. [S.l.], 1986. v. 1, p. 12. Citation on page 88.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural computation**, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citations on pages 87 and 88.

HOCHREITER, S. **Untersuchungen zu dynamischen neuronalen Netzen**. Phd Thesis (PhD Thesis) — diploma thesis, institut für informatik, lehrstuhl prof. brauer, technische universität münchen, 1991. Citation on page 86.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citation on page 86.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 2, n. 5, p. 359–366, 1989. Citation on page 87.

I CANCHO, R. F.; SOLÉ, R. V. The small world of human language. **Proceedings of the Royal Society of London B: Biological Sciences**, The Royal Society, v. 268, n. 1482, p. 2261–2265, 2001. Citation on page 78.

IACOBACCI, I.; PILEHVAR, M. T.; NAVIGLI, R. Sensembed: Learning sense embeddings for word and relational similarity. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. [S.l.: s.n.], 2015. p. 95–105. Citation on page 34.

____. 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)**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 897–907. Citations on pages 33 and 91.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; DOUZE, M.; JÉGOU, H.; MIKOLOV, T. Fasttext.zip: Compressing text classification models. **arXiv preprint arXiv:1612.03651**, 2016. Citation on page 63.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**. [S.l.]: Association for Computational Linguistics, 2017. p. 427–431. Citation on page 63.

JR, E. A. C.; LOPES, A. A.; AMANCIO, D. R. Word sense disambiguation: A complex network approach. **Information Sciences**, Elsevier, v. 442, p. 103–113, 2018. Citation on page 16.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2st. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 0130950696. Citations on pages 15 and 79.

KÅGEBÄCK, M.; JOHANSSON, F.; JOHANSSON, R.; DUBHASHI, D. Neural context embeddings for automatic discovery of word senses. In: **Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing**. [S.l.: s.n.], 2015. p. 25–32. Citation on page 33.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. Citation on page 84.

KIROS, R.; ZHU, Y.; SALAKHUTDINOV, R. R.; ZEMEL, R.; URTASUN, R.; TORRALBA, A.; FIDLER, S. Skip-thought vectors. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 3294–3302. Citations on pages 17 and 63.

KOHONEN, T. Correlation matrix memories. **IEEE transactions on computers**, IEEE, v. 100, n. 4, p. 353–359, 1972. Citation on page 82.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105. Citation on page 81.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Research, v. 521, n. 7553, p. 436–444, 2015. Citations on pages 81, 87, and 88.

LI, J.; LUONG, M.-T.; JURAFSKY, D. A hierarchical neural autoencoder for paragraphs and documents. **arXiv preprint arXiv:1506.01057**, 2015. Citation on page 91.

LING, W.; DYER, C.; BLACK, A. W.; TRANCOSO, I. Two/too simple adaptations of word2vec for syntax problems. In: **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2015. p. 1299–1304. Citation on page 63.

LUONG, M.-T.; PHAM, H.; MANNING, C. D. Effective approaches to attention-based neural machine translation. In: **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2015. p. 1412–1421. Citation on page 17.

MALLERY, J. C. Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. In: **Master's thesis, M.I.T. Political Science Department**. [S.l.: s.n.], 1988. Citation on page 17.

MANNING, C.; SCHUTZE, H. **Foundations of statistical natural language processing**. [S.l.]: MIT press, 1999. Citations on pages 15 and 19.

MARKIE, P. Rationalism vs. empiricism. In: ZALTA, E. N. (Ed.). **The Stanford Encyclopedia of Philosophy**. Summer 2015. [S.l.]: Metaphysics Research Lab, Stanford University, 2015. Citation on page 15.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943. Citation on page 82.

MIHALCEA, R.; RADEV, D. **Graph-based natural language processing and information retrieval**. [S.l.]: Cambridge university press, 2011. Citations on pages 19 and 78.

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into texts. In: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2004. Citations on pages 16 and 79.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citations on pages 16, 63, 90, and 91.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119. Citations on pages 16, 63, 90, and 91.

MILLER, G. A. Wordnet: a lexical database for english. **Communications of the ACM**, ACM, v. 38, n. 11, p. 39–41, 1995. Citation on page 78.

MITCHELL, T. M. Machine learning. 1997. **Burr Ridge, IL: McGraw Hill**, v. 45, n. 37, p. 870–877, 1997. Citation on page 81.

NARENDRA, K. S.; THATHACHAR, M. A. Learning automata-a survey. **IEEE Transactions on systems, man, and cybernetics**, IEEE, n. 4, p. 323–334, 1974. Citation on page 82.

NAVIGLI, R. Word sense disambiguation: A survey. **ACM Computing Surveys (CSUR)**, ACM, v. 41, n. 2, p. 10, 2009. Citation on page 17.

NEELAKANTAN, A.; SHANKAR, J.; PASSOS, A.; MCCALLUM, A. Efficient non-parametric estimation of multiple embeddings per word in vector space. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1059–1069. Citation on page 34.

NEWMAN, M. **Networks: An Introduction**. New York, NY, USA: Oxford University Press, Inc., 2010. ISBN 0199206651, 9780199206650. Citation on page 75.

NEWMAN, M. E. Fast algorithm for detecting community structure in networks. **Physical review E**, APS, v. 69, n. 6, p. 066133, 2004. Citation on page 78.

____. Finding community structure in networks using the eigenvectors of matrices. **Physical review E**, APS, v. 74, n. 3, p. 036104, 2006. Citation on page 77.

____. Modularity and community structure in networks. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 103, n. 23, p. 8577–8582, 2006. Citation on page 77.

NIU, Z.-Y.; JI, D.-H.; TAN, C. L. Word sense disambiguation using label propagation based semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics**. [S.l.], 2005. p. 395–402. Citation on page 79.

PARDO, T. A. S.; ANTIQUEIRA, L.; NUNES, M. d. G. V.; JR, O. N. O.; COSTA, L. da F. Modeling and evaluating summaries using complex networks. In: SPRINGER. **International Workshop on Computational Processing of the Portuguese Language**. [S.l.], 2006. p. 1–10. Citation on page 79.

PELEVINA, M.; AREFIEV, N.; BIEMANN, C.; PANCHENKO, A. Making sense of word embeddings. In: **Proceedings of the 1st Workshop on Representation Learning for NLP**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 174–183. Citation on page 34.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543. Citation on page 63.

PEROZZI, B.; AL-RFOU, R.; KULKARNI, V.; SKIENA, S. Inducing language networks from continuous space word representations. In: **Complex Networks V**. [S.l.]: Springer, 2014. p. 261–273. Citations on pages 34 and 79.

PIANTADOSI, S. T.; TILY, H.; GIBSON, E. The communicative function of ambiguity in language. **Cognition**, Elsevier, v. 122, n. 3, p. 280–291, 2012. Citations on pages 15 and 17.

QUISPE, L. V.; TOHALINO, J. A.; AMANCIO, D. R. Using virtual edges to improve the discriminability of co-occurrence text networks. **Physica A: Statistical Mechanics and its Applications**, Elsevier, p. 125344, 2020. Citation on page 48.

RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. Language models are unsupervised multitask learners. **OpenAI Blog**, v. 1, n. 8, p. 9, 2019. Citation on page 63.

RANZATO, M.; POULTNEY, C.; CHOPRA, S.; LECUN, Y. Efficient learning of sparse representations with an energy-based model. In: MIT PRESS. **Proceedings of the 19th International Conference on Neural Information Processing Systems**. [S.l.], 2007. p. 1137–1144. Citation on page 87.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citations on pages 82 and 83.

____. Principles of neurodynamics. Spartan Book, 1962. Citations on pages 82 and 83.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 1986. Citation on page 84.

RUSH, A. M.; CHOPRA, S.; WESTON, J. A neural attention model for abstractive sentence summarization. **arXiv preprint arXiv:1509.00685**, 2015. Citations on pages 16 and 91.

SANTOS, L. B. dos; JÚNIOR, E. A. C.; JR, O. N. O.; AMANCIO, D. R.; MANSUR, L. L.; ALUÍSIO, S. M. Enriching complex networks with word embeddings for detecting mild cognitive impairment from speech transcripts. In: **ACL (1)**. [S.l.: s.n.], 2017. p. 1284–1296. Citation on page 48.

SCARLINI, B.; PASINI, T.; NAVIGLI, R. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In: **AAAI**. [S.l.: s.n.], 2020. p. 8758–8765. Citation on page 34.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015. Citations on pages 82, 84, and 88.

SCHÜTZE, H. Context space. In: **Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language**. [S.l.: s.n.], 1992. p. 113–120. Citation on page 33.

SILVA, F. N.; AMANCIO, D. R.; BARDOSOVA, M.; COSTA, L. d. F.; OLIVEIRA, O. N. Using network science and text analytics to produce surveys in a scientific topic. **Journal of Informetrics**, Elsevier, v. 10, n. 2, p. 487–502, 2016. Citation on page 78.

SILVA, T. C.; AMANCIO, D. R. Word sense disambiguation via high order of learning in complex networks. **EPL (Europhysics Letters)**, IOP Publishing, v. 98, n. 5, p. 58001, 2012. Citation on page 19.

SOCHER, R.; PERELYGIN, A.; WU, J. Y.; CHUANG, J.; MANNING, C. D.; NG, A. Y.; POTTS, C. *et al.* Recursive deep models for semantic compositionality over a sentiment treebank. In: CITESEER. **Proceedings of the conference on empirical methods in natural language processing (EMNLP)**. [S.l.], 2013. v. 1631, p. 1642. Citation on page 91.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 3104–3112. Citations on pages 16, 63, 81, 87, and 91.

TEIXEIRA, A. S.; TALAGA, S.; SWANSON, T. J.; STELLA, M. Revealing semantic and emotional structure of suicide notes with cognitive network science. **arXiv preprint arXiv:2007.12053**, 2020. Citation on page 48.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 5998–6008. Citation on page 63.

VÉRONIS, J. Hyperlex: lexical cartography for information retrieval. **Computer Speech & Language**, Elsevier, v. 18, n. 3, p. 223–252, 2004. Citation on page 19.

WERBOS, P. J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, IEEE, v. 78, n. 10, p. 1550–1560, 1990. Citation on page 85.

WIDDOWS, D.; DOROW, B. A graph model for unsupervised lexical acquisition. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 19th international conference on Computational linguistics-Volume 1**. [S.l.], 2002. p. 1–7. Citation on page 79.

WIDROW, B.; HOFF, M. E. Associative storage and retrieval of digital information in networks of adaptive “neurons”. In: **Biological Prototypes and Synthetic Systems**. [S.l.]: Springer, 1962. p. 160–160. Citation on page 82.

WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. **Neural computation**, MIT Press, v. 8, n. 7, p. 1341–1390, 1996. Citation on page 92.

YANN, L. **Modèles connexionnistes de l’apprentissage**. Phd Thesis (PhD Thesis) — These de Doctorat, Universite Paris 6, 1987. Citation on page 88.

ZEILER, M. D. Adadelta: an adaptive learning rate method. **arXiv preprint arXiv:1212.5701**, 2012. Citation on page 84.

ZHONG, Z.; NG, H. T. It makes sense: A wide-coverage word sense disambiguation system for free text. In: **Proceedings of the ACL 2010 System Demonstrations**. Uppsala, Sweden: Association for Computational Linguistics, 2010. p. 78–83. Available: <https://www.aclweb.org/anthology/P10-4014>. Citation on page 19.

APPENDIX A

COMPLEX NETWORKS

The use of complex networks has become very popular in recent years, mainly due to its general purpose representation, which can be adapted and applied to different real systems (COSTA *et al.*, 2007; NEWMAN, 2010). With its origin in graph theory, a consolidated field in mathematics and computer science, the field of complex networks has as its main objective the study of networks with irregular, complex and dynamic structures, a common pattern in complex systems (BOCCALETTI *et al.*, 2006). Some of these systems come from areas such as Sociology, Communication, Biology, Physics and Economics (COSTA *et al.*, 2011), which has made this area of research highly multidisciplinary and also popular with an increasing number of researchers, including in the areas of Linguistics and Natural Language Processing.

The basic structure for the study of complex networks is called graph or network. Formally, a graph or network can be represented as an ordered pair $G = \{V, E\}$, formed by a set $V = \{v_1, v_2, \dots, v_n\}$ of vertices (or nodes, or points) and a set $E = \{e_1, e_2, \dots, e_m\}$ of edges (or connections). If it is necessary to represent the strength of the connections, the network can be defined as $G = \{V, E, W\}$, where a new set $W = \{w_1, w_2, \dots, w_m\}$ representing the weights of the edges is incorporated. Usually, graphs are represented as adjacency lists or adjacency matrices (COSTA *et al.*, 2007), in which adjacency lists are generally more space efficient. Nevertheless, for the sake of clarity in defining the network measurements in the next section, we will use the adjacency matrix as a standard representation. We can define the adjacency matrix A, where each element a_{ij} has a value of 0 or 1, where 1 represents an edge between the vertices v_i and v_j and 0 the absence of an edge. In weighted networks, the values assigned to the elements of A are not confined to 0 and 1 (they are usually continuous) and if the network is directed, a_{ij} and a_{ji} will not be equivalent, but in non-directed networks the adjacency matrix is symmetric.

After modeling a system into a complex network, it is necessary to characterize its structure, a process normally performed using topological measurements or by analyzing the overall structure of the network. In the case of measurements, created to capture specific phenomena of complex systems, they usually provide a complementary view to traditional statistical techniques, while the overall analysis of the network structure provides a macroscopic and even in some cases a mesoscopic view of the system in hand. In the following sections, we explore some commonly used network measurements (Section A.1) and also present the concept of communities in networks (Section A.2), a macro structure found in many complex systems. Finally, in Section A.3 we present how the theory of complex networks can be applied to the area of PLN, in addition to some important works related to the objectives of this work.

A.1 Characterization of complex networks

In recent years, many measurements have been developed, driven mainly by the need to characterize and describe complex networks and the complex systems they represent, respectively (COSTA *et al.*, 2007). Some of the more traditional ones are the measurements of *degree*, *clustering coefficient* and *geodesic path*. The *degree* of a vertex v_i is defined as the number of edges connected to this vertex, considering the adjacency matrix A, the *degree* can be computed by

$$k_i = \sum_{j=1}^{n} a_{ij}. \tag{A.1}$$

In directed networks, the degree can be redefined to take into account the edges that come out of a vertex (k_i^{out}) and the edges that reach a vertex (k_i^{in}). Another variation of *degree* is *strength*, a generalization for weighted networks. The calculation of *strength* is the same as the equation A.1, but considering the matrix A as a weighted adjacency matrix. With the measure of *degree* defined, it is possible to define a very simple global measure of connectivity called *average degree*:

$$\langle k \rangle = \frac{1}{|V|} \sum_{i=1}^{n} k_i. \tag{A.2}$$

The measure *clustering coefficient* is also related to network connectivity, as it represents the tendency of the network's vertices to group together. Also called transitivity, the measure *clustering coefficient* is related to the presence of triangles in the network and can be defined as:

$$C = \frac{3N_{\triangle}}{N_3}, \tag{A.3}$$

where $N_{\triangle}$ is the number of triangles in the network and N_3 is the number of connected triples. An alternative to *clustering coefficient* (C) is *local clustering coefficient (cc)*, defined as the ratio of the actual number of edges between the neighbors of the vertex i (v_i) and the maximum possible number of edges between the neighbors of vertex i (given by $k_i(k_i-1)/2$, k_i is the number of neighbors of vertex i), that is,

$$cc_i = \frac{2e_i}{k_i(k_i-1)}. \tag{A.4}$$

This measure can also be transformed into a global measure ($\langle cc \rangle$), by averaging the values of cc for all vertices in the network:

$$\langle cc \rangle = \frac{1}{|V|} \sum_{i=1}^{n} cc_i. \tag{A.5}$$

The difference between C and $\langle cc \rangle$ is that C assigns the same weight to each of the network's triangles, while $\langle cc \rangle$ assigns the same weight to each vertex of the network independently of its connectivity, resulting in different values. Still, the values of both measures and *local clustering coefficient (cc)* are between 0 and 1, where 1 represents maximum neighborhood connectivity.

The *geodesic path* or shortest path is the path between the v_i and v_j vertices with the shortest length, where the length of a path is defined by the number of edges along that path. Considering l_{ij} as the geodesic distance between the vertices v_i and v_j, we can represent all the geodesics of a graph in a matrix D, where each element $d_{ij} = l_{ij}$. From this matrix we can define the *diameter* δ as the maximum value contained in D and the *average minimum path* l of the network as

$$l = \frac{1}{|V|(|V|-1))} \sum_{i \neq j} d_{ij}. \tag{A.6}$$

This definition diverges if any of the vertices of the network is disconnected, one solution is to consider only the pairs of vertices belonging to the largest component of the network. Another alternative is to consider a related measure, such as *global efficiency*, defined by

$$E = \frac{1}{|V|(|V|-1))} \sum_{i \neq j} \frac{1}{d_{ij}}. \tag{A.7}$$

This measurement quantifies the efficiency of the network in sending information between vertices, assuming that the efficiency of sending information between two vertices is proportional to the inverse of the distance of its geodesic (COSTA *et al.*, 2007).

In general, network measurements are created in order to capture some specific phenomena in the structure of a network, which explains the growing number of measurements found in the literature. As in our work the focus is to explore the ability to learn in a network, instead of using it in the characterization process we restrict ourselves to presenting a very limited and popular group of measurements. A comprehensive review of network measures can be found at Costa *et al.* (2007).

A.2 Community structure in networks

A very common phenomenon in nature and consequently in many complex systems is the formation of groups with similar elements. In complex networks this phenomenon is defined as a community structure, i.e., there is a tendency for the vertices to divide into groups, in which the vertices of a group are highly connected to each other and sparsely connected to the other vertices of the network (GIRVAN; NEWMAN, 2002; NEWMAN, 2006a). One measurement that attempts to quantify this concept is the *modularity* (NEWMAN, 2006b), although the definition of communities is widely accepted, it is open to interpretation (how to define *highly* connected?). Thus, the *modularity* attributed to a set of partitions (of a network) is defined as the density of edges within communities compared to edges between communities. Their values are comprised between -1 and 1, where larger values indicate that the defined communities are properly communities and not structures that could be found in a random network.

The *modularity*, in addition to being used to measure the quality of communities found by different methods, is also used by several methods of community detection as an objective function for maximization. Defined as modularity maximization methods, these methods propose alternative ways to attack the maximization problem, since addressing the problem directly would have a high computational cost (NEWMAN, 2004) and prohibitive in scenarios where networks are extremely large (BLONDEL *et al.*, 2008).

An example of a modularity maximization algorithm is the Louvain method (BLONDEL *et al.*, 2008), a *greedy* algorithm that can be divided into two stages: (1) initially, the method considers each vertex to be a community and then merges the communities that would maximize the *modularity*; (2) the communities obtained are then transformed into a network, where each vertex represents a community and the edges of the new vertices correspond to the existing links between communities in the previous structure (if the network is weighted, the sum of the weights is performed). When this new structure is built, step 1 is performed again. The two steps are then repeated in each iteration of the algorithm until the communities stabilize and there is no more change in *modularity*. In addition to dividing the vertices of a network into communities, the method also provides a hierarchical structure that comes from all the iterations of the algorithm, this feature allows the analysis of the community structure with a different granularity.

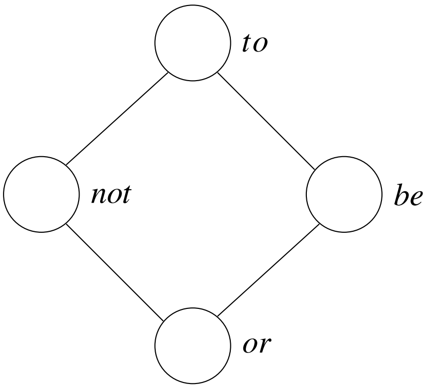
The Louvain method (BLONDEL *et al.*, 2008) was the main method used in this work, mainly because it presents some interesting properties such as low computational cost (FORTUNATO, 2010; SILVA *et al.*, 2016), a performance better than several other similar algorithms (BLONDEL *et al.*, 2008; FORTUNATO, 2010) and the possibility to use weighted networks. A more extensive review of community detection methods can be found at Fortunato (2010).

A.3 Complex networks in NLP

The use of graph theory in the area of Natural Language Processing is not a recent event, with some works dating back to the 1980s (MIHALCEA; RADEV, 2011). Nonetheless, the study of language from the perspective of a complex system and its modeling in complex networks is more recent (I CANCHO; SOLÉ, 2001). In this section we will present three popular network models used in NLP, together with some works that made their use. The three models are: semantic networks, co-occurrence networks and similarity networks.

Semantic networks are structures that represent concepts at their vertices and semantic relations at their edges. An example of a semantic network is WordNet (MILLER, 1995), a computational lexicon based on psycholinguistic principles. In WordNet, each vertex represents a set of meanings (*synsets*) and the edges represent semantic relations, such as synonyms, antonyms and hyponyms. Its use in NLP is quite popular (JURAFSKY; MARTIN, 2000), mainly in the area of word sense disambiguation, where the distance between concepts is commonly used as a semantic distance between words, the WordNet is also used by various knowledge-based disambiguation methods (AGIRRE; EDMONDS, 2007).

Figure 1 – Co-occurrence network of the phrase *"To be, or not to be"*, extracted from the book Hamlet by William Shakespeare.

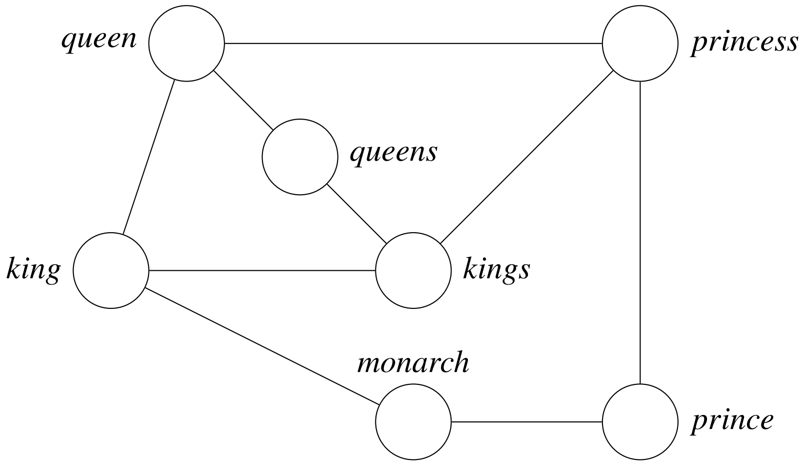


Source: Elaborated by the author.

Co-occurrence networks model the text in a structure where the vertices represent words and the edges represent co-occurrence in a window of size N, in its simplest version only adjacent words are considered, this network is sometimes said adjacency network of words. In Figure 1 we have the example of an adjacency network. This category of network has been used in several applications including part-of-speech tagging (BIEMANN, 2006), word sense disambiguation (WIDDOWS; DOROW, 2002), keyword extraction (MIHALCEA; TARAU, 2004) among others. Specifically using a characterization process based on complex networks measurements as described in Section A.1, we have works in word sense disambiguation (AMANCIO; JR; COSTA, 2012), text categorization (ARRUDA; COSTA; AMANCIO, 2016), authorship recognition (AMANCIO *et al.*, 2011) among others (AMANCIO, 2013).

Finally, we have the similarity networks, structures that represent entities at their vertices (e.g., words, sentences or paragraphs) and similarity or proximity relations at their edges. A variation of this very popular structure is the k-NN network, where each element to be modeled is connected with the k other elements closest to it. In Figure 2 we have part of a k-NN network. This category is not as popular as the others, despite its popularity in the area of complex networks (COMIN *et al.*, 2020). This model was used in applications such as summarization (PARDO *et al.*, 2006; ANTIQUEIRA *et al.*, 2009), representation of *word embeddings* (PEROZZI *et al.*, 2014), evaluation of translations (AMANCIO *et al.*, 2008) and word sense disambiguation (NIU; JI; TAN, 2005).

Figure 2 – Part of a *k*-NN network generated with *word embeddings* that were trained using Word2Vec. The parameter *k* was set to 3.



Source: Elaborated by the author.

APPENDIX B

DEEP LEARNING

Machine learning is a subfield of Artificial Intelligence (AI) that aims to build algorithms that have the ability to improve their performance on a task through experience, which is usually obtained through examples (MITCHELL, 1997). While in traditional AI approaches, known as knowledge-based AI, knowledge about the world (or problem) is encoded through manually created rules (which can later be used to make inferences and make decisions), machine learning methods on the other hand, are designed to have the ability to extract knowledge and patterns directly from the data, with very limited manual interference. This characteristic is one of the main factors of the success of methods of this nature in several areas, such as image recognition (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), automatic translation (SUTSKEVER; VINYALS; LE, 2014), speech recognition (HINTON *et al.*, 2012) and even in medical areas as in the classification of skin cancer (ESTEVA *et al.*, 2017).

Although machine learning methods perform the learning process automatically, they have a great dependence on the representation of the data, that is, in most cases the algorithms receive as input a set of features and not raw data. The features or characteristics, are sets of information that ideally have the ability to describe the data and are designed according to the nature of the problem. This process, which became known as feature engineering, was present in most of the works that involved the use of machine learning and is still a very popular phase in the design of intelligent systems today. A challenge with this approach is the difficulty of designing the features and deciding which ones to use (LECUN; BENGIO; HINTON, 2015).

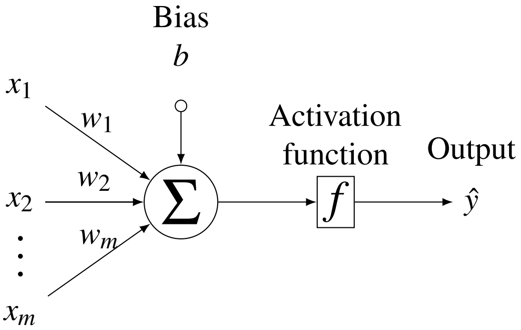
One way of tackling this problem in some tasks would be the use of representation learning instead of manually designed features. This alternative is ideal in cases where a large human effort or the effort of a entire area of research is required (GOODFELLOW; BENGIO; COURVILLE, 2016). A example of representation learning algorithm is the *autoencoder*. Better defined as a class of algorithms than a single algorithm, the autoencoders are trained in the task of reconstruction, where the model attempts to copy or map the input to the output, in this process the input information is encoded (*encoder* function) into an *intermediate representation* and later decoded (*decoder* function) to the original space (SCHMIDHUBER, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016)(more details in Section B.1). The *autoencoders* are trained to preserve as much relevant information as possible in the intermediate representation, which can later be used by any learning algorithm.

Representation learning is one of the fundamental ideas of *deep learning*, which uses deep neural networks that abstract this concept in each layer of the network, making it possible to learn complex concepts from raw data. In the next section we present the general concepts of artificial neural networks, their learning units, how learning is encoded in its internal structure, how its training is carried out, its architectures and how the area of artificial neural networks has become the popular area of *Deep Learning*. In Section B.2, we describe the main advances made by the application of deep learning to NLP tasks. We also present some recent works that try to combine complex networks and deep learning methods (Section B.3). Finally, in Section B.4 we give some remarks on the use of non-neural machine learning algorithms.

B.1 Artificial neural networks

The initial idea of artificial neural networks arose in the pioneering work of McCulloch and Pitts (1943), which presented an artificial neuron model based on the knowledge of biological neurons at the time. The proposed model had no learning capabilities, so, in order to perform some task, was necessary to manually configure the neuron. A few years later, the first work appeared that presented a form of learning (HEBB, 1949), which triggered a series of works proposing learning methods for neural networks, both supervised (ROSENBLATT, 1958; ROSENBLATT, 1962; WIDROW; HOFF, 1962; NARENDRA; THATHACHAR, 1974) and unsupervised (GROSSBERG, 1970; KOHONEN, 1972). Although the mentioned works are accepted as the starting point of the neural networks area as a machine learning method, some works emphasize that the first neural networks were basically variations of the linear regression (SCHMIDHUBER, 2015), which go back to Gauss's works in 19th century.

Figure 3 – Artificial neuron model.



Source: Adapted from Haykin (2001).

The artificial neuron, the main element of a neural network, has little relation to the biological neuron despite its name. Some of these relationships are, their ability to receive input and output (after certain processing) and the ability to be combined into more complex structures. In Figure 3 we have an example of an artificial neuron. In this structure, the entry is represented by the vector $\boldsymbol{x} = [x_1, x_2, \dots, x_m]$, the summation $\Sigma = \sum_{i=1}^{m} w_i x_i + b$ and f is an activation function that generates the network output from the summation (Equation B.1). The weights of the neuron are represented by $\boldsymbol{w} = [w_1, w_2, \dots, w_m]$ and they are responsible for the learning itself. The model also includes a *bias* b, which allows the function values to be shifted if necessary.

$$\hat{y} = f(\sum_{i=1}^{m} w_i x_i + b) \tag{B.1}$$

Several activation functions are found in the literature, some of the most popular are: sigmoid function (Equation B.2), hyperbolic tangent function (Equation B.3) and rectified linear unit (ReLU) (Equation B.4). Activation functions have a fundamental role in the generalization power of neural networks and in the learning process, as will be discussed below.

$$\hat{y} = f(x) = \frac{1}{1 + \exp(-x)} \tag{B.2}$$

$$\hat{y} = f(x) = \tanh(x) \tag{B.3}$$

$$\hat{y} = f(x) = \max(0, x) \tag{B.4}$$

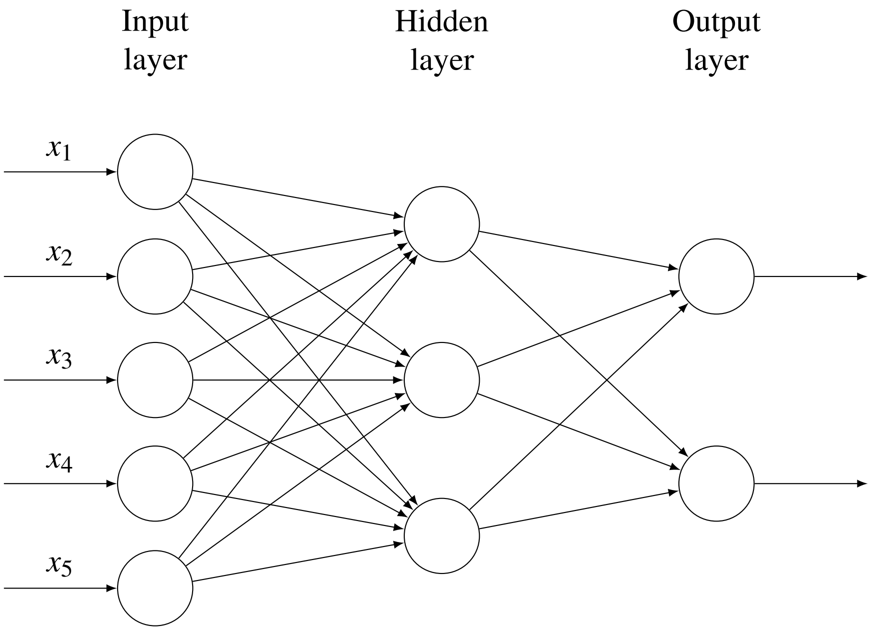
The presented neuron model has the same structure as the *perceptron* (ROSENBLATT, 1958; ROSENBLATT, 1962), the first neuron model with the capacity to learn. This model is limited to learning linear functions, but when combined in more complex structures, the so called neural network architectures, the limitation does not hold.

In a neural network architecture, neurons are connected to each other, where the output of one neuron can be connected to the input of another neuron or to the input of itself. In Figure 4 we have a example of the architecture of *feedforward* neural networks, also known as *multilayer perceptrons* (MLPs). In this structure, we have an input layer, one or more intermediate layers (hidden layers) and an output layer. The term *feedforward* refers to the fact that the flow of information flows from the network input directly to the output, with no *loops* in the structure.

In a neural network, learning is carried out in the same way as it is done in a simple neuron, by adjusting the connection weights, also known as synapses. Thus, considering a supervised learning task where it is necessary to estimate a f function that maps an entry $\boldsymbol{x}$ to a category or value y, a feedforward neural network models a function $f^*(\boldsymbol{x}; \boldsymbol{\theta})$ and learn the value of the parameters $\boldsymbol{\theta}$ (weights and *bias*) in order to better approximate f.

One of the most popular supervised learning algorithms for adjusting weights in a neural network is the *Backpropagation*, popularized by the work of Rumelhart, Hinton and Williams

Figure 4 – *Feedforward* neural network architecture



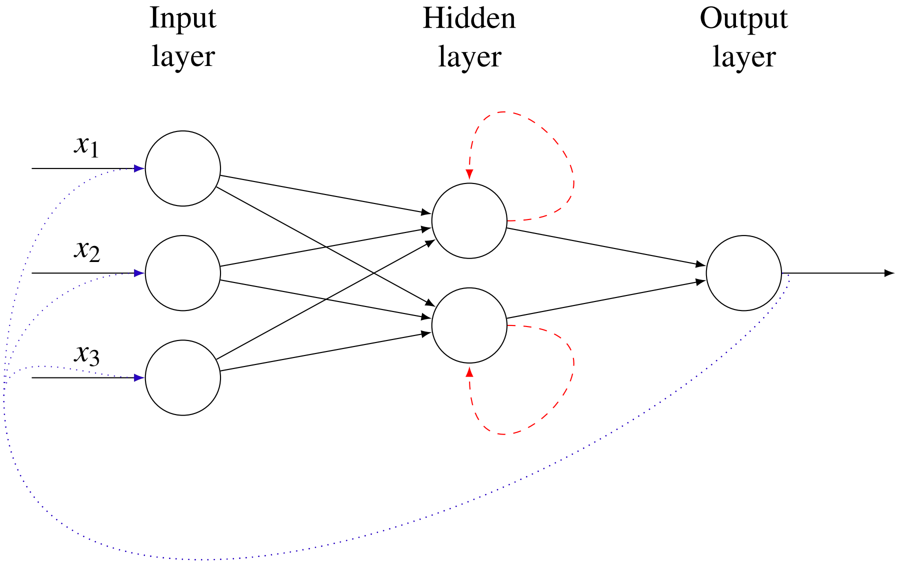
Source: Adapted from Haykin (2001).

(1986) but having its origins in several previous works dating from the 1960s (SCHMIDHUBER, 2015). In a process similar to what is done in other supervised learning algorithms, the neural network receives an input $\boldsymbol{x}$ and produces an output $\hat{y}$, from the output $\hat{y}$ it is possible to calculate the **error** in relation to the expected value (*gold*) y, the *backpropagation* then uses this error to estimate the adjustments of the network weights, the adjustments that is performed by a optimization method, such as *stochastic gradient descent* (GOODFELLOW; BENGIO; COURVILLE, 2016), AdaGrad (DUCHI; HAZAN; SINGER, 2011), AdaDelta (ZEILER, 2012) and Adam (KINGMA; BA, 2014).

Another very popular architecture is the recurrent neural network. This architecture extends the *feedforward* model by including recurrences or loops in the architecture, as can be seen in Figure 5. Created to process a sequence of values, recurring networks receive a sequence of vectors $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(\tau)}$ with τ vectors, also called steps or time steps.

Recurrent networks abstract the idea of parameter sharing and unlike what occurs in feedforward networks where a sequence would need to be modeled by different parameters (an entire sequence would feed the network), in recurring networks each element of the sequence is modeled by the same set of parameters allowing a greater generalization of the problem, because instead of trying to generalize a window of information (all vectors of the sequence), the recurring model has the ability to generalize the information distributed in a sequence, which allows this structure to model sequences that have different sizes and also to model the dependency between

Figure 5 – Recurrent neural network architecture. Connections dotted in red represent local recurrences and connections dotted in blue represent global recurrences.



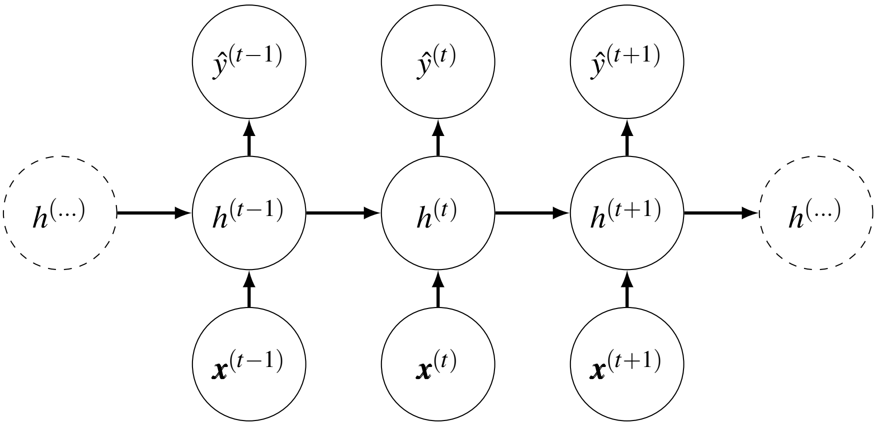
Source: Adapted from Haykin (2001).

states, because the information persists through the network recurrence.

In Figure 6 we have a state graph that represents a generalization of the functioning of a recurring network. The input sequence is represented by $(\boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t+1)})$, h represents the current state of the network (including weights, bias and other parameters) and $\hat{y}$ represents the output of the network. It is possible to notice that in a given step t, the network produces a result $\hat{y}^{(t)}$ from the entry $\boldsymbol{x}^{(t)}$ and the previous state $h^{(t-1)}$, indicating how information persistence is performed in this architecture. Another interesting feature of recurrent neural networks is their versatility in modeling different problems, for example, we can model a network that generates an output at each step (Figure 6) but we can also model a network which generates a result only at the end of an entire sequence (Figure 7), in which the first approach could be used in the translation process and the second to predict the next value in a sequence of numbers.

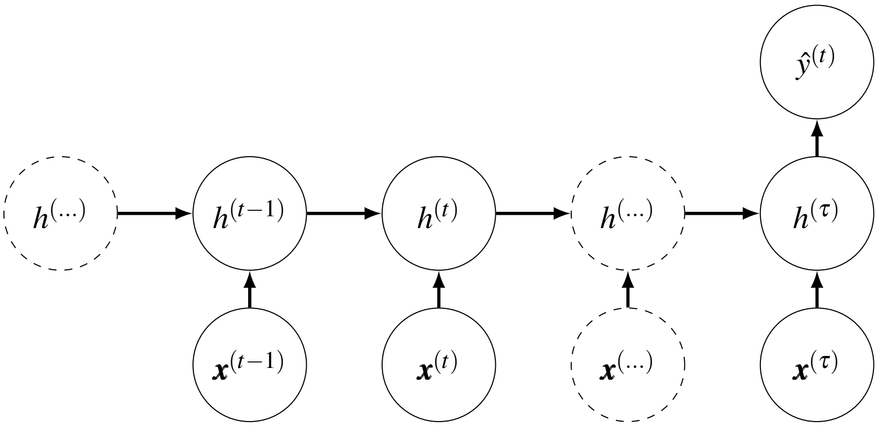
The training of this architecture is performed through a variant of Backpropagation, which has modifications to consider the time steps of a recurring network, this variant is known as Backpropagation through time (WERBOS, 1990). Thus, the combination of a recurring architecture capable of modeling sequences with the training algorithm Backpropagation through time made this type of network very popular and consequently applied in several scenarios. But despite this, the architecture presented problems in tasks that required long-term dependencies

Figure 6 – State graph of a recurrent neural network where an output is produced at each time step.



Source: Adapted from Goodfellow, Bengio and Courville (2016).

Figure 7 – State graph of a recurrent neural network where an output is produced after reading an entire sequence.



Source: Adapted from Goodfellow, Bengio and Courville (2016).

(i.e., when one element of the sequence has dependence on another distant element), this problem became known as *vanishing gradient* (HOCHREITER, 1991; BENGIO; SIMARD; FRASCONI, 1994) because the long-term dependencies of the sequence were not learned by the network and the network was unable to generalize. The vanishing gradient problem also applies to very deep feedforward neural networks.

In order to solve the problem of vanishing gradient, Hochreiter and Schmidhuber (1997) proposed a new type of recurrent neural networks, the *Long Short Term Memory networks* (LSTMs). Developed to avoid the problem of not learning long-term dependencies, these networks introduce the concept of memory cells, which are used instead of neurons. The memory cells have so-called *gates* that control the flow of information, allowing the network to learn which information must be forgotten and which information must be kept, thus learning long and short term dependencies. Similarly to the structures on Figures 6 and 7, in LSTMs, the information flows between states (e.g., from $h^{(t)}$ to $h^{(t+1)}$) are now controlled by mechanisms in the network itself.

In the literature it is possible to find several works applying LSTMs to the most diverse problems, including machine translation (SUTSKEVER; VINYALS; LE, 2014), handwriting recognition (GRAVES *et al.*, 2009), speech synthesis (FAN *et al.*, 2014). In addition to modifications in its structure to deal with specific problems (BAYER *et al.*, 2009) or simplifications of its structure to decrease the computational cost (CHO *et al.*, 2014a).

There are currently a huge number of different network architectures, some architectures are created to deal with a specific type of information, as is the case with convolutional neural networks that were created to deal with images that come in a *grid* or matrix (GOODFELLOW; BENGIO; COURVILLE, 2016). But in general, several models can be generated from simple modifications or adaptations of the presented architectures.

A very important element to be defined when creating an architecture is its *depth*, a term that refers to the number of hidden layers in the model. The universal approximation theorem (CYBENKO, 1989; HORNIK; STINCHCOMBE; WHITE, 1989) dictates that a feedforward neural network with a single hidden layer (depth 1) has the capacity to approximate any continuous function that maps a space of finite dimension to another, considering that the network has the necessary amount of neurons in the hidden layer (GOODFELLOW; BENGIO; COURVILLE, 2016). From the theorem it is possible to state that regardless of the function that needs to be learned, we only need a feedforward network large enough to represent the problem, however, there is no guarantee that the learning algorithm will be able to learn this function (the set of weights that represents that function). It is in this context that deeper models are adopted, as they can reduce the number of neurons needed to represent a function and at the same time allow a better generalization of the problem (better learning). It is also possible to interpret deeper models as being a composition of simple concepts, where the various layers of the network make the composition of simpler functions to perform a more complex function (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016).

A problem that arose at the very beginning of the adoption of deep neural networks was its training, although theoretically plausible, training networks with a large number of hidden layers did not lead to good results, until the work of Hinton, Osindero and Teh (2006). The problem in training deep neural networks is not related to a limitation of the representation, but to the limitations of the training algorithm, which gets stuck in local minima, making the network unable to adequately generalize the problem. To solve this problem Hinton, Osindero and Teh (2006) proposes the use of an unsupervised pre-training technique, which is applied to each of the hidden layers of a Deep Belief Network (type of neural network that stacks several other generative neural networks) before taking supervised training. This technique was later applied to several other types of deep neural networks (BENGIO *et al.*, 2007; RANZATO *et al.*, 2007) and in all cases the generalization of the model improved.

The unsupervised pre-training technique used by Hinton, Osindero and Teh (2006) goes back to the ideas of representation learning and distributed representation (HINTON, 1986), in which each layer of a neural network learns a new representation of the data and this new representation is fed to the next layer, which does the same procedure until the classification (or any other task) is performed on the last layer of the network. In this way, we can think that each layer is responsible for learning a set of properties, which make the classification at the end of the network possible. Hinton, Osindero and Teh (2006) took advantage of this idea by pre-training each layer of the network (starting from the input layer towards the output layer) in the reconstruction task, that is, each layer of the network received the output of the previous layer and had as a task the reconstruction of that same information. This pre-training allows the network to adjust their weights to better describe the data and consequently facilitating the supervised training that is carried out later. Considering that a network that better represents the data generates better results, it is possible to infer that the network learning process will be better if its training process starts with a set of weights favorable to the task, making it easier for the optimization algorithm to find a global minimum or a better local minima. It is also worth mentioning that since this pre-training uses only the input data, we can consider it unsupervised.

Still following the idea of representation learning, we have autoencoders, neural networks that are also trained in the task of reconstruction. Its architectures are divided into two parts, the first (*encoder*), which receives an entry $\boldsymbol{x}$ and produces an intermediate representation $\boldsymbol{x}^*$. While the second part of the architecture (*decoder*) receives $\boldsymbol{x}^*$ and produces $\boldsymbol{x}'$ output, which should be an approximation of the original value $\boldsymbol{x}$. The training is carried out in order to reduce the error between $\boldsymbol{x}'$ and $\boldsymbol{x}$ and can be carried out with common network training algorithms. The idea of autoencoder is not exactly learning how to perfectly reconstruct the data, but rather learning how to represent the data in a new way, prioritizing only essential information and properties of the input. Thus, after training, the *encoder* can be applied to the data set and the new representation can be used by any machine learning algorithm and not just neural networks. It is important to note that both the *encoder* and the *decoder* can simply be a layer of neurons up to complete neural networks with different architectures. The concept of autoencoders is not new (YANN, 1987), but became popular only after 2006, with the advent of Deep Learning.

The term Deep Learning, coined after the work of Hinton, Osindero and Teh (2006), has as its main objective to emphasize the research and use of artificial neural networks with several layers (deep representations). But it is also possible to note that the area is not only about building deep networks, but architecting an entire structure that will be able to learn and generalize highly complex tasks, which explains its success in several areas, including in the area of Natural Language Processing.

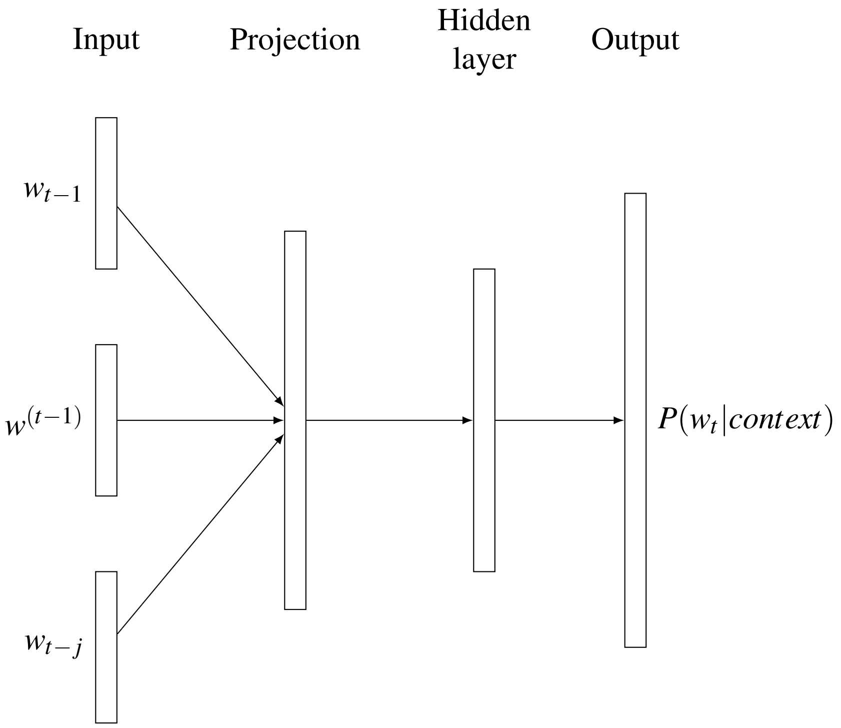
A more extensive review of deep learning, including history, methods and major advances can be found at LeCun, Bengio and Hinton (2015), Schmidhuber (2015), Goodfellow, Bengio and Courville (2016).

B.2 Deep Learning in NLP

Artificial neural networks have been used in Natural Language Processing tasks for a long time, but until the work of Bengio *et al.* (2003), its use was restricted, being considered only as another alternative of machine learning algorithm, as is case of part-of-speech tagging using the Average Perceptron (COLLINS, 2002).

Bengio *et al.* (2003) proposed a feedforward neural network architecture for the language modeling task. As the objective of this task is to build a model capable of estimating the joint probability of a sequence of words, the network was trained in the task of predicting the next word given a sequence of previous words, that is, approximating a function $P(w_t|w_{t-1},\dots,w_{t-j})$ considering a sequence of j previous words.

Figure 8 – Neural architecture for language modeling proposed by Bengio *et al.* (2003).



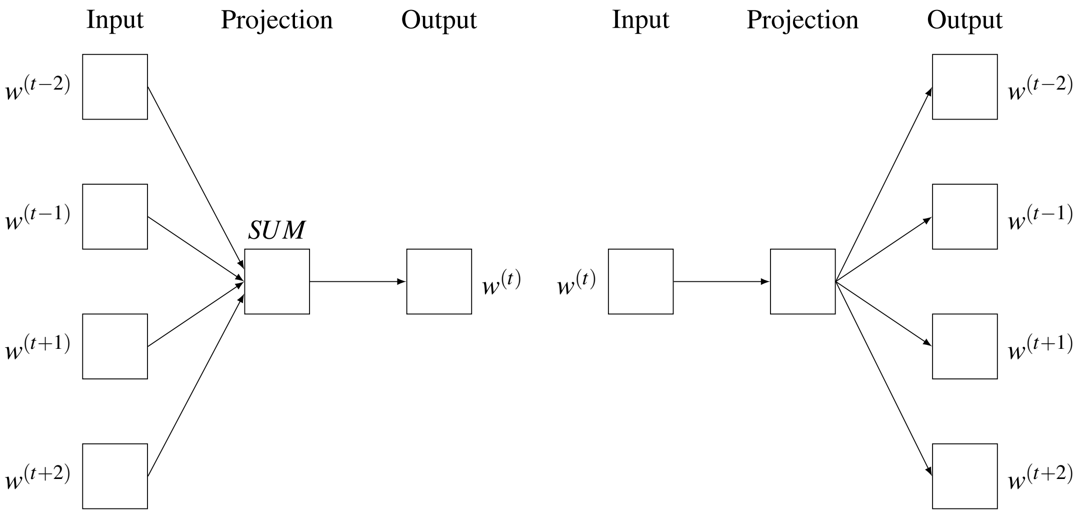
Source: Adapted from Bengio *et al.* (2003).

In Figure 8 we have the neural architecture proposed by Bengio *et al.* (2003). Each word in the sequence fed to the network is represented by a vector 1-*of*-V (*one-hot encoding*) where V is the size of the vocabulary. The input is then projected in a dense vector space of dimension D, the projection is basically the connection of each word of the input in each of the D neurons of the projection layer. The weights of these connections are defined as random at the beginning of the training and adjusted during the training like any other weight in the network. Thus, at the end of the training each word in the vocabulary will have a vector representation of dimension D, this representation is called *word embedding*.

Subsequently, the first layer of the architecture proposed by Bengio *et al.* (2003), called *embedding layer*, was incorporated by Collobert and Weston (2008) in a deeper structure and applied to several NLP tasks, including part-of-speech tagging, named-entity recognition among others. In all tasks, state-of-the-art results were obtained (COLLOBERT; WESTON, 2008; COLLOBERT *et al.*, 2011), in addition to the produced word embeddings having syntactic-semantic relationships (e.g., city name vectors were close in this space).

Although the concept of *word embeddings* was established in the works already described, it was the Word2Vec model, proposed by Mikolov *et al.* (2013a), Mikolov *et al.* (2013b) that really made the concept popular. The Word2Vec model was created with the sole purpose of generating quality word embeddings that could be used in subsequent tasks. Word2Vec is a simplified and optimized version of the model proposed by Bengio *et al.* (2003) and instead of being trained in the language modeling task, two tasks were proposed. The first being the prediction of a word given its context of neighboring words (CBOW) and the second the prediction of the context given a word (Skip-gram). In Figure 9 we have the graphical representation of the model.

Figure 9 – Simplified neural architecture proposed by Mikolov *et al.* (2013a), Mikolov *et al.* (2013b). (a) prediction of a word given its context of neighboring words (CBOW). (b) context prediction given a word (Skip-gram).



(a) CBOW (b) Skip-gram

Source: Adapted from Mikolov *et al.* (2013a).

The word embeddings produced by the Word2Vec presented the same syntactic-semantic relations found by Collobert and Weston (2008), but with a much higher quality, as indicated by the performance of the embeddings in the task of syntactic-semantic similarity (MIKOLOV *et al.*, 2013a). An interesting property of the vectors was the possibility of performing algebraic operations between vectors, such as the operation $vector('king') - vector('man') + vector('woman')$ which resulted in a vector very close to $vector('queen')$ (where $vector(x)$ is the word embeddings of the word x). This property also allowed larger blocks of information such as sentences and paragraphs to be represented by simply combining the individual vectors of each word (MIKOLOV *et al.*, 2013b). This feature made the word embeddings from Word2Vec to be used in various tasks including sentiment analysis (SOCHER *et al.*, 2013), word sense disambiguation (IACOBACCI; PILEHVAR; NAVIGLI, 2016) among other tasks (BARONI; DINU; KRUSZEWSKI, 2014).

Following the concept of word embeddings and representation learning, some models emerged with the ability to combine or encode the information of individual vectors into a single vector, that is, to represent sentences or even entire texts from the word embeddings of the input. This type of model not only allowed the generation of a compact representation, but also allowed the neural networks to operate on larger blocks of information and to directly perform tasks such as automatic translation (CHO *et al.*, 2014b; SUTSKEVER; VINYALS; LE, 2014) and summarization (RUSH; CHOPRA; WESTON, 2015).

Currently, the research of Deep Learning in NLP tries to propose both improvements and new neural models capable of performing specific tasks, as well as to propose ways of representation and acquisition of unsupervised knowledge (such as Word2Vec). An example of improvement are the *attention* mechanisms, which in a recurrent neural network means that for a step t only the most important elements prior to that step are considered, for example, during the translation of a sentence, in the production of each word in the target language only the most important words in the original sentence will be considered instead of the whole sentence (BAHDANAU; CHO; BENGIO, 2015). An example of new forms of representation are autoencoders for representing sentences or larger blocks of information (LI; LUONG; JURAFSKY, 2015; DAI; LE, 2015; GAN *et al.*, 2016).

B.3 Complex networks and deep learning

Following the success of embeddings in the NLP, the concept of *network embeddings* (GROVER; LESKOVEC, 2016; DING; LIN; ISHWAR, 2016) has become quite popular (ADHIKARI *et al.*, 2017). Ideally, methods that propose to generate *network embeddings* try to learn a set of attributes capable of representing the vertices or edges of the network in a similar way to what is done in word embeddings.

Node2Vec (GROVER; LESKOVEC, 2016) is an example of a method for generating network embeddings. This method uses the same model proposed by Mikolov *et al.* (2013a), Mikolov *et al.* (2013b) with some modifications, the words are replaced by vertices and the context (neighboring words) is defined by a random walk in the network, the rest of the algorithm remains the same. Grover and Leskovec (2016) demonstrated that the embeddings learned when used in vertex classification and links prediction obtains state-of-the-art results, that is, they can really characterize the vertices of the network. Other proposed methods follow the same idea, but with some changes in the neural structure used or in the definition of context.

B.4 Shallow classifiers

The *no free lunch theorem* for machine learning (WOLPERT, 1996) states that no machine learning algorithm is universally better than any other. This is no different for neural networks, for this reason other machine learning algorithms may be used during the work, but the choice will always be made based on the premises of the problem in question. When needed, non-neural methods will be referred to as shallow classifiers, emphasizing that these methods do not use any type of deep structure to carry out the learning.

ICMC USP
SÃO CARLOS