Word sense disambiguation: an evaluation study of semi-supervised approaches with word embeddings

Samuel Sousa
Institute of Science and Technology
Federal University of São Paulo
São José dos Campos, Brazil
samuel.bruno@unifesp.br

Evangelos Milios
Faculty of Computer Science
Dalhousie University
Halifax, Canada
eem@cs.dal.ca

Lilian Berton
Institute of Science and Technology
Federal University of São Paulo
São José dos Campos, Brazil
lberton@unifesp.br

***Abstract*—Word Sense Disambiguation (WSD) is a well-known problem in the field of Natural Language Processing (NLP) related to automatically determining the most appropriate sense of words in context. Several machine learning-based approaches have been proposed to tackle the ambiguity of language, but the lack of labeled data to train supervised models made semi-supervised learning (SSL) appear as an attractive option. Furthermore, the use of word embeddings to enhance the results of NLP tasks was shown to be an efficient strategy. Thus, this paper aims at adapting semi-supervised algorithms for WSD using word embeddings from Word2Vec, FastText, and BERT models combined with part-of-speech tags as input. We conduct a systematic evaluation of four graph-based SSL models analyzing the influence of their hyperparameters on the results, as well as the distances to build the graphs, the percentages of labeled data, and the word embeddings architectural variations. As a result, we show that SSL algorithms which received 10% of labeled data are strong baselines on the subsets of nouns and adjectives. Additionally, these algorithms do not need further training to disambiguate new words, hence being competitive to supervised systems.**

***Index Terms*—Word sense disambiguation, Semi-supervised learning, Word embeddings, Machine learning, Natural language processing, Text mining.**

I. Introduction

Words are naturally ambiguous, and their correct meanings depend on the context they are used in. Therefore, Word Sense Disambiguation (WSD) is a long-standing task of Natural Language Processing (NLP), whose definition concerns the ability to identify the accurate meaning of words in a computational manner [1]. WSD has a central role in NLP since the performance of several other tasks relies on its outcomes, such as machine translation [2], automatic text summarization [1], and question answering [3]. Nevertheless, the lack of textual data whose words are labeled with the correct meaning for both phases of training and test can be seen as the biggest barrier to develop accurate WSD systems [4]. This kind of data is expensive and time-consuming to be produced since it is manually annotated by skilled professionals from the fields of linguistics or computer science [1].

Semi-supervised learning (SSL) is a potential solution to tackle the absence of labeled data because it combines labeled and unlabeled data points in the learning process reducing the label dependency [5]. Thus, with only a small portion of labeled data, it is possible to obtain results as good as the ones achieved by supervised methods [6]. Among the SSL models, label propagation algorithms are widely used in the literature. This 'class' of algorithms uses graph structures for transductive learning [7], [8]. SSL has been successfully applied to classification [6], but the graph construction is still a challenge.

Recently, word embeddings appeared as efficient representations for words, hence they are able to keep prior knowledge which can be integrated into applied tasks [3], [4], [9]. There are several word embedding models, such as Word2Vec [10], FastText [11], and BERT [12], to name a few. Combining word embeddings with SSL for WSD is a promising technique for NLP since the vector similarity of embeddings captures the relatedness between the corresponding terms [3]. Thus, graph-based algorithms are expected to capture the patterns on the vector space of embedding models to yield structures that ease the disambiguation of words. Previous work combined SSL to augment the labeled data as the pseudo-labeling step for deep learning [4]. Other SSL works first apply a supervised method, e.g., SVM, to extract features to be used as input to Label Propagation (LP) algorithm [13]. Additionally, in most real-world scenarios, labeled data is hard to obtain and methods that can also learn from unlabeled data are highly desirable. Graph-based approaches model a function $\hat{f}(x)$ from a few labeled data instances to spread out their label information over a structure that captures the patterns from the original data set [5].

Our contributions are fourfold: 1) we first combine word embedding as features for SSL methods employing different graph-based algorithms and three-word embedding models, i.e., Word2Vec (with both CBOW and Skip-gram architectures), FastText, and BERT; 2) we perform a systematic analysis about how the parameters of word embedding (like the number of dimensions), and the parameters of graph-based algorithms (like distance functions) affect the performance of WSD; 3) we perform statistical analysis and demonstrate that our methods reach scores close to state-of-the-art supervised WSD systems; 4) the results of our experiments are the best scoring among the semi-supervised systems on most of the lexical sample benchmark data.

The remainder of this work is organized as follows. Section

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II gives an overview of related works. The detailed description of the problem, data, and work tools can be found in Section III. The experimental setup is detailed in Section IV. Section V presents the results and discussion. Finally, the conclusions and future works are presented in Section VI.

II. Related Works

Research in WSD began in the late 1940s [1] and remains an active field of NLP since numerous words are ambiguous, and computational models are not able to fully disambiguate them yet. To cope with the problem of finding the correct meanings of words in context, ML algorithms have been widely used by way of supervised [9], [14], unsupervised [15], and semi-supervised approaches [13], [16], [17]. Supervised methods [14] learn from data in which the word senses are annotated as labels, e.g., a key to the correct sense on WordNet. This approach is based on the assumption that contextual information can provide a good approximation to word meaning [9], in spite of suffering from the dependence on labeled instances for training [1]. Usually, systems based on Support Vector Machines (SVM), like IMS [14], achieve the highest scores on most of the benchmark data sets. SSL algorithms, on the other hand, present robustness on tasks that have few labeled data instances available for the learning process [5]. For WSD, the LP algorithm [7] has been frequently used [13], [16] as an alternative to supervised models.

In recent years, the performance of ML-based WSD systems has improved with the employ of word embeddings [4], [9], [18] since these word vectors capture linguistic knowledge [3]. However, classic word embedding models, such as Word2Vec, Glove, and FastText, lead to the conflation problem [15], which refers to the inability to distinguish the different meanings of a word, based on its vector representation itself [3]. The first way to tackle this embedding models drawback is done by averaging the vectors of the words near the target word for disambiguation [9], while the other strategy consists of the use of context embedding models [3]; and the second one consists in extracting vectors from models which pay attention to the whole context of a word. Context embedding models can be seen as variations of regular word embedding algorithms, which yield a vector representation for a window of words in a sentence instead of a representation for a single word [3], e.g., BERT [12]. Other knowledge features from the text, e.g., part-of-speech (POS) tags [9], [14], [18], are also exploited to enhance disambiguation results and surpass the conflation problem.

Supervised architectures, like It Makes Sense (IMS) [14], have been using word embeddings to reach state-of-the-art results on several standard data sets [9], [18], while in the SSL domain, the most known works do not make use of those prominent resources. The Local and Global Consistency (LGC) algorithm was applied by [16] over a two words data set, using only semantic and syntactic features. LP was also employed for WSD in similar purposes by [13], which performed an entropy-based feature selection and used SVM outcomes to boost its results. Recent semi-supervised models for disambiguation implement concepts from Network science, such as bipartite and multipartite networks. For instance, the IMBHN algorithm [17] presented robustness to little labeled data by the use of a bipartite structure to assign senses to ambiguous words, however, it did not employ word embeddings in the representation of the words. Among these methods, LP is the most popular algorithm, besides also being efficient for techniques of pseudo-labeling which expand training sets for deep learning-based WSD approaches [4].

III. Graph-Based Semi-Supervised WSD

In this section, we describe the problem of semi-supervised WSD, highlighting our insights to combine graph-based algorithms of SSL with word embeddings. The details of features, tools, and data sets are also provided.

A. The Problem

In the task of WSD, let W be a set of words $\{w_1, w_2, ..., w_n\}$ in which some words have their senses annotated. The set of senses is represented by S, and W can be modeled as a semi-supervised problem formulation in the form

$$W = \{(W_L, S_L), (W_U, S_U)\}, \tag{1}$$

in which W_L is the subset of words whose senses S_L are known, and W_U represents the subset of words with unknown senses S_U. Thus, SSL WSD aims to find S_U given $(W_L, S_L) + W_U$. This problem is performed as a classification task whose label to be predicted is the most adequate word sense [1]. Therefore, the set of classes is as large as the set of words, since words may present more than one sense. Moreover, semi-supervised classification setups exploit dense and sparse regions of the vector space to define the decision boundary (cluster and smoothness assumptions) and then separate the classes in the data [5].

B. WSD Features

To model the relation between words and word senses as a function $\hat{f}(x_i)$, WSD approaches commonly use a set of semantic features [1], [9] to reproduce the context of the words, combined to syntactic features which hold the values concerning POS tags and dependency relations. From the text snippets in which the target words for disambiguation are placed, we extract the following features:

- POS: We use the POS tags of all the words in a window of three words on both sides of the target word besides its own POS tag [14], [18]. If there are fewer than three words in this window, or it crosses the sentence boundaries, a null value replaces each missing POS tag [14].
- $Context_{emb}$: 10 words before and 10 words after the target word, which may include words from the prior and posterior sentences besides the target word sentence, are gathered into this feature [9]. These words have their vector representation from Word2Vec or FastText models extracted and averaged as a context embedding [3]. On the strategies for averaging, each word receives a weight to quantify its importance. Among the averaging functions,

the exponential decay strategy is the most efficient one since it assigns weights to the words exponentially in the form:

$$e_i = \sum_{j=I-W,\ j\neq 1}^{I+W} w_{ij}(1-\gamma)^{|I-j|-1}. \quad (2)$$

Where γ is the decay parameter, and w_{ij} is related to the weight associated with the i-th dimension of the j-th word in the window W of surrounding words of the target word I. In this strategy, the target word is held-out from the window of words [9]. By exponential decay, a word immediately before or after the target word has a weight 10 times higher than one placed 10 words distant. On tests with BERT vectors, this feature was not used.

- $Word_{emb}$: The vector representation of the target word for disambiguation was also used as a separate feature.

C. Word embedding Models

Word embeddings are a hot topic in NLP research and industry because of their ability to generate vector spaces in which knowledge of the language is preserved and retrievable to downstream applications [10], [11]. Aiming to analyze the effects of vectors on WSD results, we tested embeddings from two regular algorithms (Word2Vec and FastText) and a language model based on transfer learning (BERT).

Word2Vec is a model proposed by [10] in 2013. It is a shallow neural network of two layers which presents two architectures: Continuous Bag-of-Words (CBOW) and Skip-gram. The former predicts a target word by taking its context words as input, while the latter does the opposite. CBOW minimizes the following loss function

$$Loss = -log(p(\vec{w_t}|\vec{W_t})), \quad (3)$$

where w_t is the target word, and $W_t = \{w_{t-n}, \ldots, w_t, \ldots, w_{t+n}\}$ is the set of words in its context [3]. On the other hand, *Skip-gram* maximizes the average log-likelihood in the form

$$\frac{1}{N}\sum_{t=1}^{N} \sum_{-J\leq j\leq J, j\neq 0} log\, p(\vec{w_{t+j}}|\vec{w_t}), \quad (4)$$

where N is the number of training words $\{w_1, w_2, \ldots, w_N\}$, whereas $-J$ and J are the same-sized windows of words on the sides of w_t, whose index j is equals to zero [19].

FastText [11] is similar to Word2Vec, supporting both CBOW and Skip-gram architectures, besides taking into account n-grams information. This is also a two-layer neural network, which minimizes the negative log-likelihood below:

$$-\frac{1}{\Omega}\sum_{\omega=1}^{\Omega} y_\omega log(f(BA\omega)), \quad (5)$$

where Ω is a collection of documents, x_ω is the bag of features of the ω-th document after normalization, y_ω is the model outcome, A and B are two weight matrices [11].

BERT, or Bidirectional Encoder Representations from Transformers, is a deep bidirectional neural model that creates representations for language from unlabeled text [12]. BERT-based architectures have led to state-of-the-art results over a broad range of NLP tasks, such as question answering, sentiment analysis, and sequence tagging. This model can be seen as an approach for transfer learning on textual data, hence there is no need to re-train it on task-specific data [12]. The base version of BERT has twelve layers, seven hundred seventy-eight hidden sizes, twelve self-attention heads, and one hundred ten million parameters.

D. Semi-Supervised Algorithms

We have compared four SSL algorithms applied to WSD which are described as follows. The LP [7] version used in this work was implemented by [20], which takes into account the influence of neighboring vertices to determine the probability (F) of output labels as follows:

$$F_{iy} = \frac{1}{Z_i}\left(O_{iy} + \beta\sum_j A_{ij}s_{ij}(y)\right), \quad (6)$$

where O_{ik} is the likelihood the vertex i has the label y, $s_{sj}(k)$ is a signal from i to vertex j that means the strength by which i believes j has the label y; Z_i is a normalization term to ensure $F(iy)$ sums to 1 over all elements whose label is y; A_{ij} is a ij-element of adjacency matrix A; and β is the strength parameter in the interval $(0, \infty]$.

LGC [8] exploits the smoothness between both labeled and unlabeled data instances. To predict the label of the unlabeled points, the following iteration equation is used:

$$T^r = \alpha\mathcal{L}T^{r-1} + (1-\alpha)I \quad (7)$$

where T^r corresponds to a stochastic matrix at the r-th iteration; α is the clamping factor parameter in range (0, 1); $\mathcal{L}$ is the graph Laplacian; and I represents the identity matrix.

Gaussian Random Fields (GRF) [21], which implements the concept of harmonic functions to predict the labels for the unlabeled subset, has the form

$$\widehat{g}(v_j) = \frac{1}{D_j}\sum_{i\sim j} P_{ij}\widehat{g}(v_i), \quad (8)$$

in which the label of the vertex v_i is assigned according to the sum of the weights of its neighboring nodes v_j; D is the diagonal degree matrix; and P is the affinity matrix.

OMNI-Prop (OMNI) [22] considers that each vertex has two scores: a self-score q_{iy} and a follower score δ_{iy}. The former holds the likelihood of vertex i to hold the label k, while the latter holds the likelihood of the neighborhood of i to present this same label. It works by iteratively updating these scores in the form

$$q_{iy} = \frac{\sum_{j=1}^{n} A_{ij}\delta_{jy} + \lambda b_y}{\sum_{j=1}^{n} A_{ij} + \lambda} \quad (9)$$

then,

$$\delta_{jy} = \frac{\sum_{i=1}^{n} A_{ij}q_{iy} + \lambda b_y}{\sum_{i=1}^{n} A_{iy} + \lambda} \quad (10)$$

where Aij is the ij-element in the adjacency matrix; b_y in each equation is the prior score; and λ is the prior strength parameter, which controls the updates of b_y [22].

E. The Data Sets

TABLE I
LEXICAL SAMPLE WSD DATA SETS STATISTICS

Data set	Size	Number of Word Senses	Number of Target Words
Senseval-2 LS	13,093	889	73
Senseval-3 LS	11,804	323	57
Semeval-2007 LS	27,125	368	100
Semcor	226,040	30,242	23,341

Lexical sample (LS) data sets for WSD were released by discovery challenges in which a disambiguation task was proposed and are widely used to rank disambiguation models in the literature. Their structure encompasses training and test subsets, with a greater number of data instances when compared to all-words WSD benchmarks. Each data instance comprises a target word for disambiguation within a text snippet extracted from large corpora. Semcor [23], otherwise, is the largest data set for WSD manually annotated and freely available on the Internet. It has no subset division and it is commonly used as training data for supervised models [9]. We have applied SSL algorithms over **Senseval-2 LS** [24], **Senseval-3 LS** [25], **Semeval-2007 LS** [26], and **Semcor** data sets, whose statistics are shown in the Table I.

IV. EXPERIMENTAL SETUP

From the pre-processing of the data sets to statistical tests and validation of results, we performed 5 steps, which are depicted in Figure 1 and described below.

Step 1. To pre-process the benchmark data sets, we have tokenized, lemmatized, removed the stopwords, POS-tagged, and further selected: ten words on the sides of the target word; three POS tags likewise the window of ten words; and the target word itself enclosed to its POS tag.

Step 2. Embeddings from Word2Vec, FastText, and BERT were extracted to represent the words for disambiguation in the data sets. Firstly, we trained the Word2Vec model with CBOW and Skip-gram architectures on the English Wikipedia corpus varying the dimensions in $\{250, 500, 1000\}$, hence yielding six different models whose vocabularies comprise the 2,787,545 most frequent words in the corpus. The remaining embedding models used in this study were pre-trained and released by their authors. We used 300-dimensional FastText vectors made available by Facebook Research and the pre-trained BERT model from Pytorch.

Step 3. After extracting the embeddings and combining them with the POS tags, we constructed graphs for running the SSL algorithms of LP, LGC, GRF, and OMNI using four popular distance measures for text mining tasks (Cosine, Euclidean, Manhattan, and Chebyshev), in order to discover which one is the best to represent the clusters related to word senses in the vector spaces. All graphs for SSL were constructed with the k-nearest neighbors (k-NN) technique whose parameter k assumed values in $\{1, ..., 10, 15, 20, 25, 50, 75, 100\}$. In this case, each word represented by its set of features (Section III-B) is connected with the k closest/most similar words.

Step 4. The LP, LGC, GRF, and OMNI algorithms were run over the built graphs. Their hyperparameters (α, β, λ) had optimum values reached by grid search technique, and the ratio of labeled data was also varied in $\{25\%, 50\%, 75\%, 100\%\}$. Each trial was repeated 30 times with random sampling for selecting the labeled subset.

Step 5. In order to evaluate the SSL algorithms performance, we have computed the F1 score values for each trial and then averaged them at the end of the 30th execution. The standard deviation was also calculated to be input to statistical tests to validate our results.

V. RESULTS AND DISCUSSION

This section outlines the experimental results of our methods on LS benchmarks, analyzes the distance measures used for graph construction, and shows the influence of the embedding models on the final performance for WSD. In order to ease the tables reading, some symbols and acronyms are used according to the following definitions. The letter 'k' in parentheses is the number of nearest neighbors assigned to construct the graph. Greek letters α, β, and λ stand for the SSL methods' hyperparameters. Furthermore, the names of cosine, euclidean, Manhattan, Chebyshev distance metrics were shortened respectively to: "*Co.*", "*Eu.*", "*Ma.*", and "*Ch.*".

A. Lexical sample WSD Results

With 250-dimensional CBOW and Skip-gram vectors, 300-dimensional FastText embeddings, and BERT vectors with 768 dimensions, we have evaluated the SSL algorithms on the three LS benchmark data sets. In Table II, the best F1 scores of our SSL models are presented. These scores were reached when the original training subsets were totally input to the models.

When the major word sense is assigned to all occurrences of its target word, the most frequent sense (MFS) baseline appears [1]. All distance measures beat MFS. In general, the LP algorithm achieved the highest F1 scores. LGC is sensitive to the presence of noise in the data, which, in this case, are the words out of the vocabularies of Word2Vec and FastText models which were replaced by a vector of the same dimensionality filled out with zeroes. OMNI suffers from the major class influence, i.e., the word sense assigned to most of the words in the labeled subset, which rises the values of sense probability to this class. Hence specific words disambiguation is poorly performed, as showed in Table II. The harmonic functions implemented by GRF are robust to noise, as well as the lack of hyperparameters. This method ranked at the first place on Semeval-2007 LS data set, with Skip-gram embeddings, and on Senseval-2 LS with BERT embeddings.

In our experiments, Skip-gram gives higher results compared to the CBOW model because it better represents infrequent words since it does not make use of a loss function based

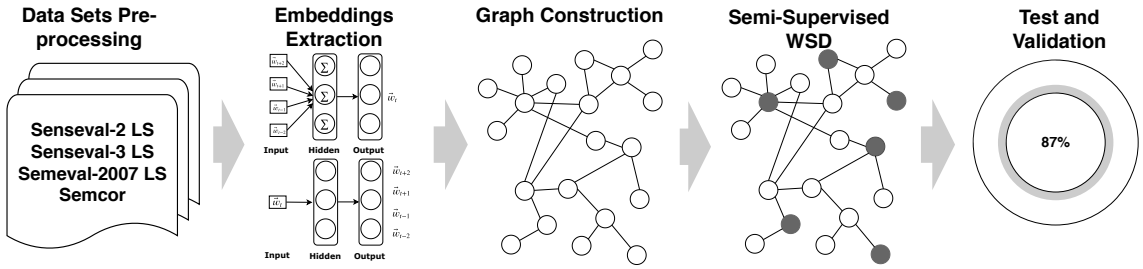


Fig. 1. Process pipeline with five steps from the pre-processing of WSD benchmark data sets to evaluation of SSL WSD performance.

TABLE II
F1 PERFORMANCES ON THE THREE SENSEVAL AND SEMEVAL LS BENCHMARKS

	CBOW			SKIP-GRAM			FASTTEXT			BERT		
Model	**SE-2**	**SE-3**	**SE-7**	**SE-2**	**SE-3**	**SE-7**	**SE-2**	**SE-3**	**SE-7**	**SE-2**	**SE-3**	**SE-7**
LP *Co.* (k=5, β=0.25)	**57.2**	66.9	**83.7**	**59.3**	**68.8**	85.2	**59.7**	**69.1**	**85.5**	68.4	**73.8**	86.9
LP *Eu.* (k=5, β=0.25)	**57.2**	66.8	**83.8**	**59.2**	**68.7**	85.2	**59.5**	**69.1**	**85.5**	68.6	**73.3**	87.1
LP *Ma.* (k=5, β=0.25)	57.0	66.8	**83.8**	59.1	**68.6**	85.2	**59.6**	69.0	**85.5**	68.2	**73.3**	87.2
LP *Ch.* (k=6, β=0.25)	56.0	65.1	82.8	55.2	65.2	83.4	55.8	64.6	83.6	63.5	65.1	84.0
LGC *Co.* (k=15, α=0.25)	56.7	**67.5**	79.9	58.8	68.5	85.1	59.0	**69.8**	**85.5**	68.1	73.1	**87.3**
LGC *Eu.* (k=15, α=0.25)	56.7	**67.5**	79.9	58.8	68.4	85.1	58.7	68.9	**85.5**	68.0	73.2	87.1
LGC *Ma.* (k=10, α=0.25)	**57.4**	**68.3**	83.2	**59.2**	68.4	**85.3**	58.8	68.9	**85.5**	68.1	73.2	**87.5**
LGC *Ch.* (k=4, α=0.75)	55.8	64.9	82.1	55.2	65.2	83.4	55.7	62.8	83.7	63.8	65.1	84.3
GRF *Co.* (k=6)	56.0	66.7	83.0	58.8	68.4	**85.3**	59.3	68.4	85.4	**68.7**	72.9	87.2
GRF *Eu.* (k=7)	56.0	66.7	83.0	58.8	68.4	**85.3**	59.0	68.4	85.4	**68.7**	72.6	87.1
GRF *Ma.* (k=5)	56.0	67.0	83.2	58.3	68.1	85.2	58.0	68.7	85.4	**68.7**	72.6	**87.4**
GRF *Ch.* (k=5)	54.9	64.1	82.1	54.2	65.2	83.6	53.9	64.7	81.6	64.4	64.8	84.8
OMNI *Co.* (k=4, λ=1.0)	55.3	61.8	82.6	57.6	67.7	84.5	57.0	67.9	83.2	67.8	72.1	84.7
OMNI *Eu.* (k=3, λ=1.0)	55.2	61.3	82.4	57.3	67.1	84.6	57.1	68.2	83.3	67.7	71.8	85.3
OMNI *Ma.* (k=5, λ=1.0)	55.2	62.0	82.6	57.4	67.7	84.4	57.1	67.4	83.0	67.9	72.1	85.1
OMNI *Ch.* (k=6, λ=1.0)	54.7	60.4	81.5	54.0	63.4	82.3	54.5	62.8	81.4	63.6	68.9	82.5

on probability. The greatest difference between the scores on embeddings of these architecture models is noticed on the results of OMNI with Cosine distance on the Senseval-3 LS data set. Skip-gram embeddings led to a 9.5% higher F1 score. On all data sets, Skip-gram embeddings were more efficient for WSD, since the textual domains of the benchmark data sets (general news) are similar to Wikipedia articles that we used to train the Word2Vec model. Furthermore, when FastText vectors were used, the difference to Skip-gram vectors was slight, since both models share some properties. The results of our experiments suggest that both models perform equivalently good disambiguation. BERT embeddings, on the other hand, surpassed the performance of all the other embedding models tested. Considering its architecture model which is based on bidirectional transformers and able to differentiate the usages of words and benefit finer meanings which used to be conflated to the most frequent ones by the other embedding methods. BERT also provides satisfactory results even though only 10% of the original training subset is used as discussed later in this subsection.

In Table III the SSL algorithms are compared against their supervised and semi-supervised counterparts. The WSD results achieved by other SSL baselines [13], [16], [17] and supervised state-of-the-art systems [9], [14], [18], [27] are listed. Our results surpassed IMBHN [17] by at least 8.17% and 2.5% of F1 score on Senseval-3 LS and Semeval-2007 LS data sets respectively when Skip-gram, FastText or BERT embeddings were used. This algorithm used a bipartite network to spread word senses in an SSL way, besides being the latest approach of SSL for WSD in the literature. LP, LGC, and GRF have hit the highest scores among the semi-supervised models on Senseval-2 LS and Semeval-2007 LS data sets. Moreover, our results are competitive with supervised models, since LP + BERT hit an F1 score about 1% below IMS + Word2Vec on Senseval-2, at the same time this system only surpasses LGC and GRF on Semeval-2007 by 2.2%.

Few sense-annotated corpora are available to build, train, and evaluate supervised models [1]. Therefore, we have run our algorithms on the Senseval-2 LS data set using small amounts of its original training subset to show their robustness and efficiency. Besides 25%, 50%, 75%, and 100%, we split the training subset into 10% and 90% with random selection. The remaining data instances which were not selected were added to the test subset and used as unlabeled data to measure the

TABLE III
COMPARISON OF F1 PERFORMANCES OF SSL ALGORITHMS WITH EMBEDDINGS AGAINST THE SSL AND SUPERVISED BASELINES

Model	SE-2	SE-3	SE-7
Supervised baselines			
IMS (2010) [14]	65.3	72.9	**87.9**
Taghipour and Ng (2015) [18]	66.2	73.4	–
AutoExtend (2015) [27]	66.5	**73.6**	–
IMS + Word2Vec (2016) [9]	**69.9**	**75.2**	**89.4**
Our approaches			
LP + CBOW *Eu.* (k=5, β=0.25)	57.2	66.8	83.8
LGC + CBOW *Ma.* (k=10, α=0.25)	57.4	68.3	83.2
GRF + CBOW *Ma.* (k=5)	56.0	67.0	83.2
OMNI + CBOW *Co.* (k=4, λ=1.0)	55.3	61.8	82.6
LP + Skip-Gram *Co.* (k=5, β=0.25)	59.3	68.8	85.2
LGC + Skip-Gram *Ma.* (k=10, α=0.25)	59.2	68.4	85.3
GRF + Skip-Gram *Eu.* (k=7)	58.8	68.4	85.3
OMNI + Skip-Gram *Co.* (k=4, λ=1.0)	57.6	67.7	84.5
LP + FastText *Co.* (k=5, β=0.25)	59.7	69.1	85.5
LGC + FastText *Co.* (k=15, β=0.25)	59.0	69.8	85.5
GRF + FastText *Co.* (k=6)	59.3	68.4	85.4
OMNI + FastText *Eu.* (k=3, λ=1.0)	57.1	68.2	83.3
LP + BERT *Co.* (k=5, β=0.25)	68.4	**73.8**	86.9
LP + BERT *Eu.* (k=5,β=0.25)	**68.6**	73.3	87.1
LGC + BERT *Ma.* (k=10, α=0.25)	68.1	73.2	**87.5**
GRF + BERT *Ma.* (k=5)	**68.7**	72.6	87.4
OMNI + BERT *Ma.* (k=5, λ=1.0)	67.9	72.1	85.1
Semi-supervised baselines			
LP + SVM *Cosine* (2007) [13]	–	71.7	–
LP *Jensen-Shannon* (2005) [16]	–	70.3	–
LP *Cosine* (2005) [16]	–	68.4	–
IMBHN (2018) [17]	–	63.6	83.2
MFS Baseline	47.6	55.2	78.0

TABLE IV
F1 PERFORMANCE COMPARISON OF OUR SEMI-SUPERVISED AGAINST SVM ON THE POS TAG SUBSETS OF SENSEVAL-2 LS WITH BERT EMBEDDINGS

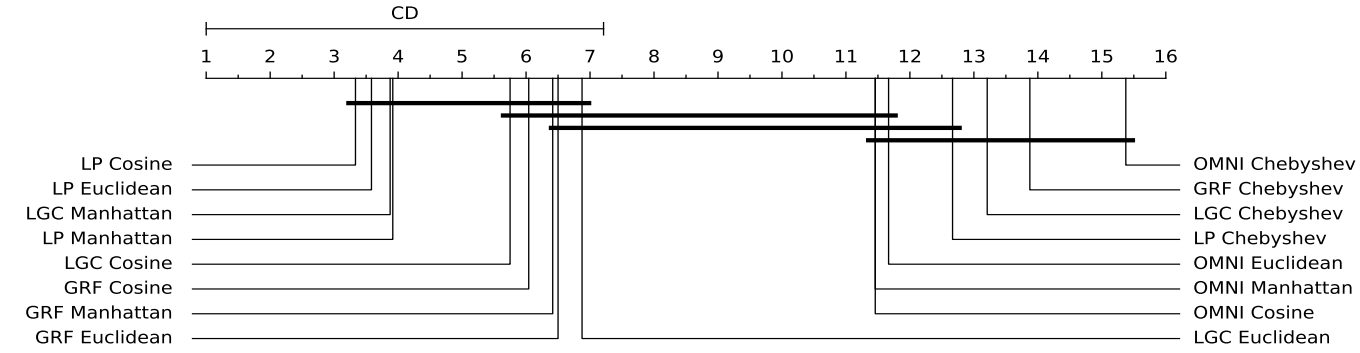
POS Tag	Model	Ratio of Labels					
		10%	25%	50%	75%	90%	100%
ADJ	SVM	74.2	81.1	82.6	80.5	**87.6**	**91.7**
	LP	80.9	83.6	84.7	85.1	85.1	86.1
	LGC	79.9	82.8	84.2	84.8	85.8	86.8
	GRF	**81.6**	84.0	**85.6**	**86.2**	86.0	86.3
	OMNI	81.2	**85.1**	84.8	84.9	85.7	85.8
NOUN	SVM	56.0	59.7	59.8	61.3	72.4	**73.9**
	LP	61.8	69.8	72.0	**72.7**	73.0	73.0
	LGC	60.9	68.5	71.0	71.8	71.8	72.4
	GRF	**61.3**	**70.4**	**72.1**	72.5	**73.2**	73.4
	OMNI	61.2	69.5	70.9	71.4	72.2	72.7
VERB	SVM	41.7	48.8	52.5	54.6	**65.6**	**67.6**
	LP	50.1	55.7	61.4	62.6	63.1	63.6
	LGC	47.7	53.5	59.1	61.1	62.8	63.3
	GRF	51.4	**56.7**	**61.9**	**63.0**	64.5	64.7
	OMNI	**51.8**	56.4	60.8	62.6	62.3	62.8
ALL	SVM	49.5	54.1	57.1	58.1	**69.9**	**70.4**
	LP	60.5	63.7	67.1	67.5	67.9	68.4
	LGC	58.7	61.6	65.3	66.0	67.1	68.1
	GRF	**61.6**	**64.7**	**67.7**	**67.8**	68.2	68.3
	OMNI	61.4	64.4	66.8	67.2	67.7	68.0

F1 score on. Furthermore, we tested our approaches against an SVM classifier with linear kernel for each POS tag subset in this data set (adjectives, nouns, and verbs) and the results can be seen in Table IV. It is possible to notice that for all POS tags, SVM only surpasses the SSL models when at least 90% of the original labeled data was used. In addition, 10% of labeled data is enough to guarantee about 80% of F1 score for adjectives and over 60% of F1 score for nouns and all POS. By analyzing the results, it is possible to notice GRF is the semi-supervised method that presents the most robustness to few labeled data instances.

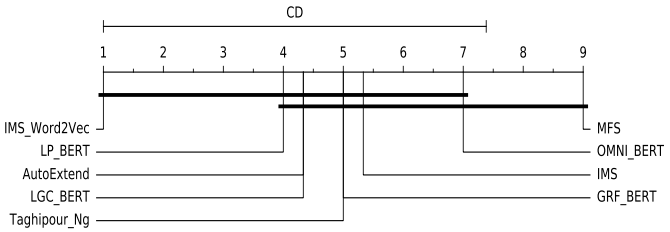
Figure 2 presents the results of the statistical analysis using the Nemenyi post-hoc test for the supervised and semi-supervised methods. The critical difference (CD) is plotted on the top of the diagrams and the average ranks of the methods are plotted in the axis, where the lowest (best) positions are on the left side. If a set of methods has no significant difference, they are connected by a black line in the diagram. For Friedman and Nemenyi statistical tests, we considered the statistics at 95 percentile. In Figure 2a, the results in Table II are tested, and the critical value of the F-statistics with 15 and 11 degrees of freedom at 95 percentile is 1.43, which the null-hypothesis of similar behavior is then rejected. The critical value for comparing the average ranking of two different methods is 6.21, and LP with cosine distance is suggested to be the most efficient method for WSD with word embeddings, according to Nemenyi post-hoc test. In Figure 2b, we compare the scores of our approaches with BERT embeddings against supervised and state-of-the-art results, the critical value of the F-statistics with 8 and 16 degrees of freedom at 95 percentile is 2.59, which rejects the null-hypothesis of similar behavior. The critical value for comparing the mean-ranking of two different algorithms is 6.94. In Figure 2c, our SSL algorithms with BERT vectors are tested against other semi-supervised baselines, and the critical value of the F-statistics with 8 and 16 degrees of freedom at 95 percentile is 2.59, which has rejected the null-hypothesis of similar behavior.According to the Nemenyi test, all label propagation algorithms combined with BERT can be statistically similar with supervised and semi-supervised baselines and the best ranked among the supervised ones are LP and LGC, while the best ranked among semi-supervised methods from the literature is GRF.

B. The influence of distance measures

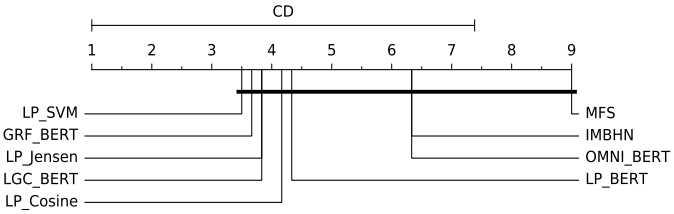
In Figure 3, the distance matrices for each distance measures are plotted. The darkest shades in each matrix show distances close to zero, whereas the lightest ones depict the opposite. Between all of these four metrics, Cosine distance (Figure 3a) led to best scoring tests on most of the setups. Chebyshev (Figure 3d) distances, otherwise, led to the lowest results, whereas both Euclidean and Manhattan distances (Figure 3b and Figure 3c respectively) presented satisfactory results.



(a) Comparison of our four SSL algorithms with their distance metrics against each other

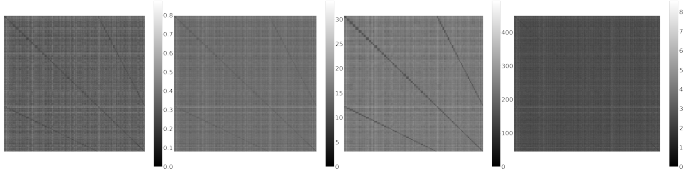


(b) Comparison of our SSL combined with BERT against supervised baselines



(c) Comparison our SSL combined with BERT against SSL baselines

Fig. 2. Statistical results with Nemenyi post-hoc test.



(a) Cosine (b) Euclidean (c) Manhattan (d) Chebyshev

Fig. 3. Distance matrices for Senseval-3 LS data set using 250-dimensional Skip-gram vectors.

TABLE V
F1 PERFORMANCES ON THE SEMCOR SAMPLE WITH VECTORS OF DIFFERENT DIMENSIONALITIES

	CBOW			SKIP-GRAM			FastText	BERT
Model	**250**	**500**	**1000**	**250**	**500**	**1000**	**300**	**768**
LP *Co.*	81.7	82.2	**82.6**	**82.8**	82.4	82.0	**84.2**	**85.3**
LP *Eu.*	81.7	**82.6**	81.9	82.3	**82.8**	**82.5**	83.9	**85.2**
LP *Ma.*	81.8	**82.5**	**82.5**	82.3	82.6	**82.6**	**84.1**	**85.2**
LP *Ch.*	79.1	80.6	81.3	81.2	80.6	81.5	82.5	83.7
LGC *Co.*	80.9	82.0	81.9	82.0	81.6	81.8	83.0	84.8
LGC *Eu.*	81.0	81.5	81.5	81.9	81.9	81.5	84.0	84.8
LGC *Ma.*	81.0	81.7	81.9	81.9	81.9	81.7	82.8	84.0
LGC *Ch.*	78.4	80.1	81.5	80.5	80.6	80.5	82.3	83.8
GRF *Co.*	81.0	82.3	82.5	**82.5**	82.3	81.5	81.4	84.9
GRF *Eu.*	81.0	82.1	82.4	82.2	81.5	82.2	81.6	84.9
GRF *Ma.*	81.0	82.4	82.4	82.3	82.3	82.1	81.5	84.8
GRF *Ch.*	78.5	81.2	81.2	80.8	81.0	81.0	82.3	83.5
OMNI *Co.*	**82.1**	81.5	82.1	81.9	82.4	81.8	83.3	84.7
OMNI *Eu.*	**82.0**	81.4	81.5	82.1	**82.8**	80.9	83.7	84.0
OMNI *Ma.*	**82.0**	82.1	81.4	82.0	82.6	81.7	83.7	84.1
OMNI *Ch.*	80.1	80.1	81.2	80.4	80.6	80.4	82.2	83.0

C. The influence of embeddings dimensionality

In order to measure the influence of parameters of embedding models (dimensionality and architecture) on WSD performance in a fair setup, a sample of 18,300 words and 183 word senses was randomly extracted from the Semcor data set. To be used as a labeled subset, we have randomly separated 2/3 of this sample, leaving 1/3 as an unlabeled subset in which the performance was measured. The pipeline for these experiments was the same as done for the LS data sets. Table V presents the F1 score for each SLL algorithm run on the Semcor sample. All results in this table were obtained by using 25% of the labeled subset, k=15, β=0.1, α=0.99, and λ=1.0. It is possible to notice that this small amount of labeled data was enough to achieve an F1 score of over 80% on almost all trials. LP was robust to the increasing dimensionality number, mainly on CBOW embeddings. OMNI benefited from 250 CBOW vector with cosine, euclidean, and Manhattan distances to reach the highest scores on these vectors. Additionally, Manhattan distance was more efficient on CBOW, whilst Euclidean appeared on half of the best scoring trials on Skip-gram vectors. FastText and BERT vectors surpassed the performance of Word2Vec regardless of presenting higher dimensionalities. Lower dimensional vectors, although, are useful to reduce the computational cost of the algorithms, since the distance metrics are computed by each vector dimension.

VI. Conclusion and Future Works

In this paper, we have tested different combinations of word embeddings, distance measures, and SSL methods applied to the problem of WSD. Our experiments were conducted on

four popular data sets for this task. A systematic analysis of graph construction, hyperparameter search, distance metrics comparison and variation of the amount of labeled data was carried out leading to the following findings. SSL algorithms can perform WSD as efficiently as supervised models with the advantage of predicting senses of out-of-vocabulary words without re-training the models as the supervised ones do. Furthermore, our results ranked first among the semi-supervised systems on most of the LS benchmark data sets, being tight to state-of-the-art supervised scores.

Among the Word2Vec architectures, Skip-gram is the most efficient one for LS WSD, since it is not based on a loss function dependent on probabilities [3], yielding meaningful representations for infrequent words. On a perfectly balanced sample of the Semcor data set, the influence of embedding parameters was noticed mainly on LP performance, which demonstrated to be robust to the increase of the number of word vector dimensions. Moreover, the difference between Skip-gram and FastText results was not large, suggesting that both models are recommended to imbalanced textual data. On the other hand, BERT is the most effective model to extract embeddings for WSD. Its efficiency can be noticed even though 10% of labels are input to the algorithms. Semi-supervised algorithms are particularly strong baselines for adjectives and nouns. Verbs, otherwise, are tough even for supervised classifiers. Among the distance measures, cosine distance presented most of the best results on the trials. In contrast, the main drawback of this work concerns the data sets since the distribution of the number of data instances per word sense is highly skewed. Finally, as future works, we plan to test new features from textual data and to apply similarity measures derived from word embedding models properly.

VII. Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001; São Paulo Research Foundation (FAPESP) grants 2018/09465-0 and 2018/01722-3; and the Natural Sciences and Engineering Research Council of Canada (NSERC). We also would like to express our gratitude to Dr. Jeannette Janssen for her insights on word sense disambiguation, BSc. Willian Dihanster for his help with the SSL algorithms, and Dr. Didier Vega-Oliveros for helping to run some experiments.

References

[1] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.

[2] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017, pp. 162–167.

[3] J. Camacho-Collados and M. T. Pilehvar, "From word to sense embeddings: A survey on vector representations of meaning," *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, 2018.

[4] D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf, "Semi-supervised word sense disambiguation with neural models," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

[5] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.

[6] L. Berton, A. de Andrade Lopes, and D. A. Vega-Oliveros, "A comparison of graph construction methods for semi-supervised learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[7] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Technical Report CMU-CALD-02-107, 2002.

[8] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, 2003, pp. 321–328.

[9] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proceedings of the 54th ACL (Volume 1: Long Papers)*, 2016.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013*, 2013, pp. 1–12.

[11] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "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*, 2017, pp. 427–431.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[13] A. Alexandrescu and K. Kirchhoff, "Data-driven graph construction for semi-supervised graph-based learning in NLP," in *NAACL-HLT*, 2007, pp. 204–211.

[14] Z. Zhong and H. T. Ng, "H.t.: It makes sense: A wide-coverage word sense disambiguation system for free text," in *In: Proceedings of the 48th ACL*, 2010, pp. 78–83.

[15] M. Pelevina, N. Arefiev, C. Biemann, and A. Panchenko, "Making sense of word embeddings," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016.

[16] Z.-Y. Niu, D. Ji, C.-L. Tan, and L. Yang, "Word sense disambiguation by semi-supervised learning," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2005, pp. 238–241.

[17] E. A. Correa Jr, A. A. Lopes, and D. R. Amancio, "Word sense disambiguation: A complex network approach," *Information Sciences*, vol. 442, pp. 103–113, 2018.

[18] K. Taghipour and H. T. Ng, "Semi-supervised word sense disambiguation using word embeddings in general and specific domains," in *Proceedings of the 2015 NAACL-HLT*, 2015.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, 2013, pp. 3111–3119.

[20] Y. Yamaguchi, C. Faloutsos, and H. Kitagawa, "CAMLP: confidence-aware modulated label propagation," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, 2016, pp. 513–521.

[21] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, 2003, pp. 912–919.

[22] Y. Yamaguchi, C. Faloutsos, and H. Kitagawa, "Omni-prop: Seamless node classification on arbitrary label correlation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 3122–3128.

[23] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *Proceedings of the workshop on Human Language Technology*, 1994, pp. 240–243.

[24] P. Edmonds and S. Cotton, "Senseval-2: Overview," in *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, jul 2001.

[25] R. Mihalcea, T. Chklovski, and A. Kilgarriff, "The senseval-3 english lexical sample task," in *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.

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

[27] S. Rothe and H. Schütze, "AutoExtend: Extending word embeddings to embeddings for synsets and lexemes," in *Proceedings of the 53rd Annual Meeting of the ACL*, Beijing, China, 2015.