Expert Systems With Applications 181 (2021) 115162

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Word sense induction using leader-follower clustering of automatically generated lexical substitutes

Abbas Akkasi [a,*], Jan Snajder [b]

[a] Department of Computer Engineering, Bandar Abbas Branch, Islamic Azad University, Bandar Abbas, Iran
[b] TakeLab, Faculty of Electrical Engineering and Computing, University of Zagreb Unska 3, 10000 Zagreb, Croatia

ARTICLE INFO

Keywords:
Word sense induction
Natural language processing
Graph clustering
Clustering refinement
Lexical substitution

ABSTRACT

Word Sense Induction (WSI) concerns the automatic identification of the various senses of polysemous words. Any improvement in this process can directly affect the quality of the applications in which knowing the word's senses is important. For example, word sense disambiguation, information retrieval, and clustering of web search result in lexically ambiguous queries. In this paper, we propose a novel WSI model that makes use of automatically generated lexical substitutes for a target word to construct a graph and data preparation for the next steps. Following the data preparation step, we make use of Leader–Follower graph clustering to find the basic senses of the target word. The senses of the target word inside the remaining or new upcoming instances will be decided according to their contextual embedding's similarities with the basic sense. Besides, to make the number of found sense groups of a target word much closer to the reality, we apply post-processing at the end. The results of experiments on SemEval2010 dataset confirm that the proposed method outperforms all the state-of-the-art solutions in terms of both harmonic and geometric v-measure and f-score with a lower average number of sense groups.

1. Introduction

Word Sense Induction (WSI) refers to automatically discovering of the word senses (meanings) from a text (Manandhar, Klapaftis, Dligach, & Pradhan, 2010). It is an open problem in the domain of natural language processing (NLP) whose solution can considerably affect many other NLP tasks. For example, web information retrieval systems on highly ambiguous queries (Véronis, 2004), or diversification of the search results returned by search engines (Navigli & Crisafulli, 2010). The WSI has also been used for enriching lexical resources such as WordNet (Nasiruddin, Schwab, Tchechmedjiev, Sérasset, & Blanchon, 2014; Miller, 1995) and increasing the quality of machine translation (Zhang, 2014). In contrast to word sense disambiguation (WSD) as a similar problem in NLP that requires a large amount of sense-labeled target words, WSI doesn't need the annotated training data (Nasiruddin, 2013). Data annotation usually is a labor expensive process for NLP tasks. Moreover, the meanings of the new words that have been added to the language or vocabulary, or novel senses of existing words cannot be discovered within the WSD process, while induction of new senses for a word is one of the main goals of the WSI.

Regardless of the diversity of the solutions to address the WSI, the existing methods are categorized into three different approaches in which the clustering is the basis for most of them: (1) Word or context clustering, (2) Lexical substitution-based methods, (3) Probabilistic methods (Nasiruddin, 2013; Turney & Pantel, 2010; Brody & Lapata, 2009; Widdows & Dorow, 2002; Alagić, Šnajder, & Padó, 2018). Clustering is the predominant approach to address the WSI problem. We also approached the problem in this context.

In general, clustering can be done in two different ways, vector-space and graph clustering. In the first approach, data points are represented as a feature vector while graph clustering algorithms work on the weighted undirected graph to find the individual sub-graph showing different clusters. Graph clustering has also been used successfully in different NLP tasks (Das, Das, Nayak, Pelusi, & Ding, 2019; Takano et al., 2019; Mills & Bourbakis, 2013; Ngomo & Schumacher, 2009).

Aside from which clustering approach to use, one basic problem that confront all the algorithms is the need to know the number of clusters in advance. This value can directly affect the performance of the WSI methods. An alternative is to make use of non-parametric clustering methods such as the *Chinese Whispers Algorithm* (Biemann, 2006), *Leader–Follower* (Shah & Zaman, 2010), and *Affinity Propagation* (Dueck, 2009), to name a few.

* Corresponding author.
E-mail addresses: abbas.akkasi@gmail.com (A. Akkasi), jan.snajder@fer.hr (J. Snajder).

https://doi.org/10.1016/j.eswa.2021.115162
Received 13 January 2020; Received in revised form 17 March 2021; Accepted 3 May 2021
Available online 8 May 2021

Nevertheless, graph based WSI methods usually require a significant amount of computational resources. To overcome the limitations and make graph based WSI more practical, we propose a simple, yet effective clustering based solution that first construct a graph making use of automatically generated lexical substitutes and contextual words of a target words inside the small portion of its instances. Leader–Follower (LF) algorithm does not need to know the number of clusters in advance, which is why we use it to find the basic sense groups. In the following, the remaining target words would be assigned to the basic groups. New sense groups would also be created if a target word does not resemble any of the previous existing cluster representatives.

It can be assumed that the use of lexical substitutions of the target word and its contextual information could improve the efficiency of the WSI system.

The proposed model can be used in an online setting to induce the sense of new words i.e. the words which are not seen in the dataset used for making the graph. New sense groups can also be created if the similarity between an instance integration vector and existing sense groups falls under a predefined threshold. The study of older WSI systems (Manandhar et al., 2010) generating an unrealistic number of meaning groups for the target word is an issue that manifests itself in most of them. To deal with this problem, post-processing is done at the end. Affinity propagation and K-means are two algorithms that we used in this stage.

In the course of the experiments, having access to the standard data which designed for the WSI task plays an important role. All experiments were thus carried out on the dataset published by SemEval2010 for the WSI task. Based on the findings, our method outperforms all of the state-of-the-art models developed using the same dataset. The rest of this paper is organized as follows: Section two and three gives a brief overview of the basic concept and related studies on WSI respectively. In section four, we describe the proposed method in more detail. The experimental setup is described in section five and results of the study are presented in section six. We also discuss the effects of post-processing on the results in the same section. Finally, we conclude the paper in section seven.

2. Background

This section aims to explain the various concepts in the WSI domain and those used within the proposed model.

Word sense induction is a challenging task of NLP whose aim is to identify and categorize different senses of polysemous target words. This process must be carried out without the help of external sense inventories like WordNet (Miller, 1998). For example, the word **race** may implies several senses depends on the context which it is appeared like: *any competition, a contest of speed, a taxonomic group that is a division of a species, etc.*

WSI methods are mostly unsupervised in which the output is a clustering of contexts in which the target word occurs or a clustering of words related to the target word. Clustering-based approaches need to have access to a massive corpus including a diverse set of instances and can be categorized into three different groups: *Word, Context and Co-occurrence graph clustering* that essentially adopt the distributional semantic approach (Turney & Pantel, 2010) based on the distributional hypothesis stating that: words are semantically similar if they appear in similar documents (Harris, 1954). *Word clustering based* methods mostly are focused on grouping the semantically similar words that may refer to a specific sense. Solutions in these approach directly leverage the word embedding produced by distributional semantics models.

In *context clustering*, a target word in each instance is represented by its context vector, then those vectors are grouped into different clusters, each corresponding to a different sense of the target word.

Co-occurrence graph clustering, relies on the idea that the meaning of a word can be represented utilizing its co-occurrence graph. The vertices and edges of such a graph are the words and the strengths of their co-occurrence associations respectively. This approach is very similar to the word clustering where the weight of edges can be obtained based on grammatical or collocational relations (Widdows & Dorow, 2002).

Most of the developed solutions for WSI were approached the problem using vector-space clustering models where either the target word or its context is represented as a feature vector (Pedersen, 2007; Manandhar et al., 2010). The constructed feature vectors are then clustered using different hierarchical or partitional clustering algorithms such as k-means, FCM [1], DBSCAN [2], to name a few (Xu & Wunsch, 2005). However, it is also possible to take advantage of the graph clustering approach (Di Marco & Navigli, 2013; Klapaftis & Manandhar, 2008; Klapaftis & Manandhar, 2010; Hope & Keller, 2013; Navigli & Lapata, 2010). In graph clustering, words or instances are considered as nodes of a graph. The edges' weight can be calculated in different ways according to the existence of a specific kind of relations between corresponding nodes in an enormous corpus. The results obtained from most recent methods are strongly depend on several parameters such as a prior number of clusters (granularity factor) or size of the input graph (Dorow & Widdows, 2003). Chinese Whisper (CW) (Biemann, 2006) is a graph clustering algorithm which it showed excellent performance on a variety of NLP tasks. Trying the **Leader–Follower (LF) graph clustering** algorithm during the experiments, we observed that LF could outperform CW.

LF is a non-parametric clustering algorithm designed to detect the communities in a graph data. It is inspired by the innate characteristics of the social networks in terms of the natural internal structure expected for the communities in such networks. LF uses the notion of distance centrality of nodes in a graph to separate the *Leaders*- the nodes which connect different communities- from the *Followers* that are the nodes with neighbors from only one community. The algorithm consists of two steps: (1) differentiation between the leader and the follower nodes (2) assigning the followers to the leaders.

Making use of the centrality concept in the graph theory, a node is a leader if its distance centrality (i.e., measures how a node is close to all other nodes in a given graph) is less than one of its neighbors; otherwise, it is a follower node. Let $d(u,v)$ denotes the shortest path distance between nodes u and v in a graph G. Then, the distance centrality $D(u)$ of $u \in V$ is: $D(u) = \sum_{v \in G} d(u,v)$.

Assuming V is the set of all nodes and L is the set of leaders, $F = V - L$ as a set of followers, and $N(v)$ gives the set of neighbors of node v, the follower assignment process is as follows:

1. Order the leader nodes per increasing distance centrality, $\{\nu_1, \nu_2, \cdots, \nu_{|L|}\}$
2. Define $M : V \rightarrow V \cup \{\star\}$ with $M(\nu) = \nu$ for all $\nu \in L$; $M(\nu) = \star$ for all $\nu \in F$.
3. For $i = 1,2,3,\cdots,|L|$, in mentioned order, define $F_{\nu_i} = \{\nu \in N(\nu_i) \cap F : M(\nu) = \star\}$; for all $\nu \in F_{\nu_i}$, set $M(\nu) = \nu_i$ and $C_{\nu_i} = \{\nu_i\} \cup F_{\nu_i}$.

It is proven that the LF can find all small scaled community structures in dense networks (Shah & Zaman, 2010). For each cluster, we select a sentence that has the highest edges' weight's sum within a cluster as the representative of that cluster. Although the graph created for WSI problem is not as dense as graph resulted by social networks, comparing to the other clustering algorithms shows the improved performance.

Lexical substitution is another concept that is used here mostly to compute the graph's edges' weights. Lexical substitution is the task of finding a set of alternatives for a word while preserving its meaning in the given context (McCarthy & Navigli, 2007; McCarthy & Navigli, 2009). For example, given the word *match* in the sentence "A red card early in the match increases the odds of winning", valid substitutes would include "game" and "race", but not "light" or "pair", even though

[1] Fuzzy C-means clustering

[2] Density-based spatial clustering of applications with noise

the latter two words are synonymous with *match* in other contexts. The manual creation of lexical substitutes is an expensive task. Furthermore, it would require human experts in the loop, which needs to have access to the instances of the target word, and thus the processing of newly coming instances would increase the costs even further. A word instance is defined as a target word along with its context i.e the sentence in which the target word appears.

We make use of lexical substitutes for instance representation. To eliminate the problems related to manual creation of lexical substitutes (LS), we utilize five state-of-the-art automatic LS generation systems proposed byMelamud, Levy, and Dagan (2015, 2016). The performance of state-of-the-art LS systems still does not match the human, thus making use of diverse LS generators theoretically should provide further benefits to instance representation. All five LS systems are essentially based on neural word embedding for instance representation.

Word2vec (Mikolov, Chen, Corrado, & Dean, 2013), is the most popular word embedding model, which learns and embeds two representations for each word: target word embedding and its context embedding. However, context representation is only used by the model only internally, and it is discarded after training. Thus, the resulting word embeddings are completely context in-sensitive. All the experiments have been done in this research basically make use of word2vec pretrained model.

The LS systems make explicit use of word2vec's context embeddings in combination with the target word embedding to construct the context-sensitive representations. Equipped with such representations, these systems select a proper substitute for a target word in a given instance by considering the similarities between the embeddings of substitute word and embeddings of both target word and its context. Four of LS systems make use of different approaches (Formulas 1–4) to measure the semantic similarity between the target word and a candidate substitute. The difference between the systems is only about the way they combine the similarity of the substitute with the target word embedding. In all schemes the compatibility of the substitute word with the context is also taken into account in terms of the similarity between their embeddings.

$$\text{Add}: \quad \frac{\cos(s,t)+\sum_{n\in C}\cos(s,c)}{|C+1|} \tag{1}$$

$$\text{BalAdd}: \quad \frac{|C|\cdot\cos(s,t)+\sum_{c\in C}\cos(s,c)}{2\cdot|C|} \tag{2}$$

$$\text{Mult}: \quad \sqrt[|C|+1]{pcos(s,t)\cdot\prod_{c\in C}pcos(s,t)} \tag{3}$$

$$\text{BalMult}: \quad \sqrt[2\cdot|C|]{pcos(s,t)^{|C|}\cdot\prod_{c\in C}pcos(s,t)} \tag{4}$$

In formulas 1–4, C is the set of the context words around the target word t, while *cos* and *pcos* are the cosine similarity and cosine similarity restricted to positive values, respectively.

The remained LS system is *context2vec* (Melamud et al., 2016)[3] which is an unsupervised model for learning the context embeddings trained with a BiLSTM (Graves & Schmidhuber, 2005). This model trained with a very large corpus and optimized to reflect the dependencies between the target words and their contextual words. We make use of pretrained word and context embeddings with all LS systems which are trained on ukWac (Ferraresi, Zanchetta, Baroni, & Bernardini, 2008) corpus.

LS systems create different sets of lexical substitutes for each target word in a small portion of instances. Then, we create an undirected weighted graph making use of substitutes' set as input for the LF algorithm. Additionally, the extracted lexical substitutes will be used for the second step of the model to assign remaining instances into the basic groups of senses or to create the news.

Although, making use of word2vec leads to the state-of-the-art results, we replaced it with fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017) and ELMo (Peters et al., 2018) to address the problems of oov (out of vocabulary) and context in-sensitivity of the word2vec respectively. fasText is an extension of the word2vec that represents each word as n-gram of characters instead of learning word vectors directly. It allows the embedding to understand the meaning of prefixes and suffixes. fastText works well with rarely used words. For oov word, it can be decomposed into n-grams to get their embedding into which word2vec does not provide any word representation for them. ELMo is another word embedding system that takes character-level tokens as input to a bi-directional LSTM (Hochreiter & Schmidhuber, 1997) to generate word-level embedding. Those embeddings are context sensitive, producing different vectors for words with the same spelling but different meanings like *bank* in *river bank* and *bank balance*.

3. The related works

The most popular approaches for WSI are usually variants of the context clustering approach introduced by Schütze (1998). In this way, the context in each instance of a target word is clustered into word senses using corresponding individual first or second-order distributional vectors (Korkontzelos & Manandhar, 2010; Hope & Keller, 2013).

HERMIT (Jurgens & Stevens, 2010), the winner solution for the WSI task in SemeEval2010, modeled the individual contexts in a high-dimensional word space and induced the word senses by finding similar contexts. Korkontzelos and Manandhar (2010), introduced an unsupervised graph based method in which the vertices of the graph represent the unambiguous units, including single words or a pair of words. The edges of the graph demonstrate the co-occurrences of the nodes that they join. AdaGram, proposed by Bartunov, Kondrashkin, Osokin, and Vetrov (2016), uses different prototypes to represent a word depending on the context to handle the various forms of word ambiguity. Mitra et al. (2014), proposed a new unsupervised method to identify noun sense changes based on rigorous analysis of time-varying text data available in the form of millions of digitized books. They constructed distributional thesauri-based networks from data at different time points and cluster each of them separately to obtain word-centric sense clusters corresponding to the different time points.

Brody and Lapata (2009), introduced a Bayesian approach for sense induction. They formulated a problem in a generative approach that describes how the contexts surrounding an ambiguous word might be generated based on latent variables. The model leverages features based on lexical information, parts of speech, and dependencies in a principled manner. The extended version of the Bayesian model for WSI proposed by Charniak (2013) incorporates the idea that the words closer to the target word are more relevant for predicting its sense. These models are of probabilistic solutions for WSI in which the context of a target word is modeled as samples from a multinomial distribution over senses.

Recent work on WSI has recognized the potential of lexical substitutes for representing word senses. Baskaya, Sert, Cirik, and Yuret (2013), proposed a system that creates a substitute vector for each target word from the most likely substitutes suggested by a statistical language model. Target Word samples are taken according to probabilities of these substitutes, and the results of the co-occurrence model are clustered. Their model assumes the generated substitutes only as an intermediate representation and does not make the conceptual link to the lexical substitutions. Moreover, they used the same number of senses for all target words. Alagić et al. (2018), investigated the use of an alternate instance representation based on the replacement of the target words with contextually suitable, meaning-preserving substitutes. They used a state-of-the-art lexical substitution method (Melamud et al., 2015; Melamud et al., 2016) to generate lexical substitutes automatically for every target word in each instance. Then, they applied a simple

[3] https://github.com/orenmel/context2vec.

clustering algorithm (affinity propagation) to induce the word senses. A recent method proposed by Amrami and Goldberg (2019), clusters lexical substitutes created by pretrained recurrent neural network (RNN) language models, specifically using contextual word representations derived from deep RNN models ELMo (Peters et al., 2018b) and BERT (Devlin, Chang, Lee, & Toutanova, 2018).

MaxMax by Hope and Keller (2013), is a non-parameterized method that finds the number of senses in a graph automatically by identifying root vertices of maximal quasi-strongly connected sub-graphs, computed in linear time. Başkaya and Jurgens (2016), proposed a new heterogeneous ensemble WSI method using the outputs of different WSI systems. Cocos and Callison-Burch (2016), introduced a similar approach for clustering paraphrases as word senses; they made use of both spectral and hierarchical clustering algorithms for the experiments carried out on their dataset. Lau, Cook, McCarthy, Newman, and Baldwin (2012), applied topic modeling based on the hierarchical Dirichlet process. Most earlier approaches to WSI are summarized in (Navigli, 2009). A recent approach by Ramprasad and Maddox (2019), explores the possibility of obtaining multi word-sense representations and sense induction to embedding spaces by jointly grounding contextualized sense representations learned from sense-tagged corpora and word embedding to a knowledge base. They integrate ontological information along with inducing polysemy to predefined embedding spaces without the need for re-training. This method is a mixture of context-based and lexical substitution-based approaches and does not rely on external resources such as WordNet. The system designed by Alagić et al. (2018), is the most similar method to the one proposed in this paper. They combine the context-based representations with automatically generated lexical substitutes. In contrast to their method, however, our method applies clustering only on a small subset of instances to detect of the underlying sense groups, and it is not necessary to have access to all the instances of a target word up front to induce its senses. Moreover, our method outperforms theirs, while yielding a lower number of sense clusters which is more close to the reality.

4. Proposed method

We propose a method consisting of three stages to address the WSI problem as illustrated in Fig. 1. There are multiple steps for each stage.

4.1. Discovery of basic senses

This is the first stage that basic sense groups of a target word is going to be discovered. In order to make use of the less amount of the instances of a target word for graph construction, firstly, a small sub set of instances is randomly sampled for each target word. Then, to utilize the LF, we create a graph $G^4 = (V^5, E^6)$ with only sub-sampled instances of a target word. Taking advantage of the idea presented in Biemann and Nygaard (2010) we regarded each instance as a node rather than words as nodes. After normalizing the instances in lower case, different sets of lexical substitutions for the target word will be generated using the automatic LS systems explained in Section 2. The size of the intersection between the sets of substitutes produced for the target words in each pair of nodes is regarded as of weight for the corresponding edge.

After constructing a weighted undirected graph, we apply the LF algorithm to produce the ”base clusters” corresponding to the main senses of the target word[7].

[4] Graph
[5] Vertices
[6] Edges
[7] We use the term ”main sense” to denote the meanings of a word that would typically be listed in a dictionary or thesaurus (to distinguish it from subsenses, sense modulations, connotations, etc.), although we acknowledge that the notion of sense is rather arbitrary.

4.2. Sense assignment

The remaining instances of the target word which are not contributed in the graph construction process, are assigned to the underlying clusters discovered before. The context-substitutional similarity between the instances and the representatives of the clusters is the key for decision making. A new instance is assigned to a cluster where it is most similar to its representative. If the highest similarity score is less than a predefined threshold, a new sense group will be created for that instance. This, later constraint guaranties that the new senses of the target word within a completely different context will also be induced.

Instance embedding here is a combination of two embedding vectors:

1. **Context:** Motivated by the basic principle of distributional semantic - *you shall know a word by the company it keeps* - (Firth, 1935) and robustness of word embedding usages in NLP problems. We construct the context embedding vector of an instance by averaging the word embeddings of the content words surrounded the target word within the given instance (Formula 5).
2. **Substitutes:** We create a substitute embedding vector by averaging the word embedding vectors of all individual substitutes generated by LS systems for a target word in a given instance (Formula 6).

A combination of the context and substitutes embeddings is used as the final embedding vector for every instance of a target word by simply taking their average. For each pair(instance - sense's representative), the cosine similarity (Formula 7) used as a score to decide which group of senses the instance should be assigned to.

$$ContextEmbd = \frac{\sum_{tok \in \{Tokens\ \ of\ \ the\ \ instance\}} WE(tok)}{L_t} \tag{5}$$

$$SubstituteEmbd = \frac{\sum_{Sub \in \{Generated\ \ substitutes\}} WE(Sub)}{L_s} \tag{6}$$

$$Sim(Ins,\ Rep) = c_c * CosineSim(ContextEmbd\_Ins,\ ContextEmbd\_Rep) + c_s * CosineSim(SubstituteEmbd\_Ins,\ SubstituteEmbd\_Rep) \tag{7}$$

In the above formulas *WE* is the pretrained model that returns the word embedding of its input parameter if it exists. L_t and L_s are the number of the tokens and substitutes generated that are not out of vocabulary in the WE model respectively. *WE* can be of traditional static models like Word2vec or Glove, etc. Although, the efficacy of recent *WE* models has been demonstrated in the following section, it is not necessary to modify the structure of our solution to use these type of WE models. A similarity score between an instance (Inst) and a cluster representative (Rep) is a weighted sum of their context and substitutional similarities with c_c and c_s coefficients respectively. Those values indicate the importance of the contextual and substitutional information in assigning a sense to the target word respectively. In the experimental phase, they were also regarded as 0.5.

4.3. Post-processing

The target word within sentences with different syntactic structures may lead to the creation of a strange instance embedding. Such unusual embedding sometimes results in numerous clusters with only one or two instances. To deal with this problem and bring the results much closer to reality, we apply some post-processing at the end. In reviewing the induced clusters for a target word, it was realized that there are clusters with only one or two cases as their members. In order to reduce the average number of clusters while maintaining overall performance, we tested different techniques for this purpose. First, for single-member clusters, instances are re-assigned to other multi member clusters

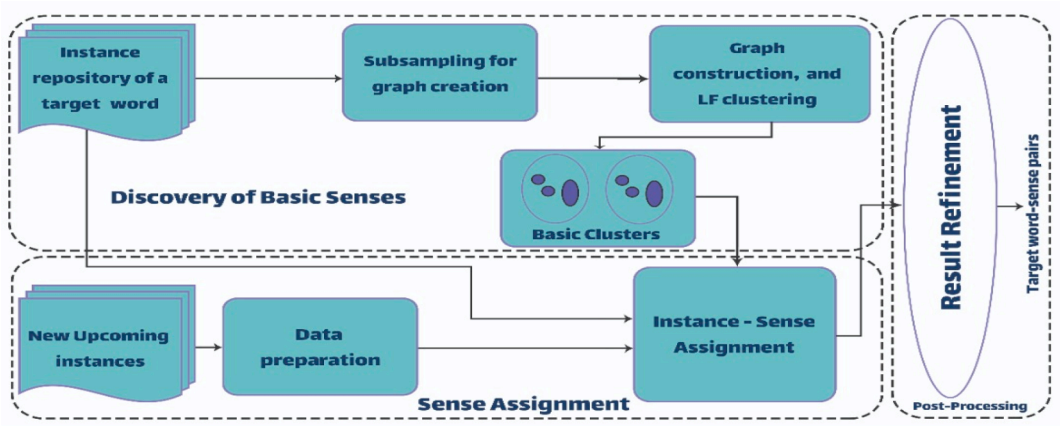

Fig. 1. Three main steps of the proposed WSI method: (1) initial clustering for obtaining the "basic senses", (2) assignment of new word instances to the basic senses, and (3) results refinement(post-processing).

based on their similarities to cluster representatives. Of course, at this stage, we relax the similarity threshold slightly less than when we apply it to the primary process. We also tried to re-cluster these singleton cluster instances using two commonly used algorithms i.e. Affinity propagation, and K-means, respectively. For Affinity propagation,it is not necessary to have preliminary information on the number of clusters, but for K-means that it is necessary. Moreover, following the main idea in this paper in using the automatically generated lexical substitutes along with the context of a target word, we applied the same idea to create feature vectors to feed into the two aforementioned algorithms.

As described in previous sections, the method proposed in this research uses the Leader–Follower approach as the primary clustering algorithm. In "sense assignment" step, what we do is not a clustering from the scratch. We only map the remaining instances to clusters based on how similar they are to clusters' representatives. But in post-processing, we apply the affinity-propagation and k-means to reduce the number of induced clusters. In this step, we focus on the clusters with only one or two members and do clustering from the scratch on those clusters' members. We would therefore argue that the clustering methods in Section 4.3 cannot be used in Section 4.2.

5. Experimental setup

We conducted the experiments using the test data released by SemEval2010 for the task of WSI. Moreover, in order to optimize the hyper-parameters, we used the trial dataset of the SemEval2013 graded WSI task (Jurgens & Klapaftis, 2013). In the following subsections, we first introduce both datasets used for the experiments in detail. The evaluation measurements and the process we followed to fine-tune the different hyper-parameters of the model are discussed next.

5.1. Used datasets

To compare our proposed method with the previous works, we use the dataset from the SemEval2010 released for the WSI task. The dataset includes 100 target words such that half of them are nouns and the rest are verbs. Each target word comes along with a fixed set of training and test instances as context. The training data were collected using the web; hence they were highly noisy. The testing dataset is part of the OntoNotes (Hovy, Marcus, Palmer, Ramshaw, & Weischedel, 2006) coming from various news sources including the Wall Street Journal, CNN, ABC and the others. The presented model in this paper only uses the context of a target words inside the corresponding instances and considering that our aim at comparing performance with the previous systems, we only used test data for our experiments.

We also leveraged dataset provided by Erk, McCarthy, and Gaylord (2009) to fine-tune the hyper-parameters. This dataset includes 8 target words with 50 instances per each one. Target words are annotated using WordNet 3.0 senses by three annotators. The annotators provided a rate between 1–5 for each sense indicating the suitability of the sense according to the given context. To avoid using test data to fine-tune the hyper-parameters, we tried this small independent dataset. Since, in this work we were focused on single sense extraction, we simply took the sense with highest total rank as a single gold sense of a target word in given instance.

Both test and trial datasets we used are summarized in Table 1.

5.2. Fine-tuning the model's hyper-parameters

There are some hyper-parameters for each stage of the proposed model that must be properly initialized. Table 2 shows them all and their corresponding values which we have examined in our experiments with trial data. To find the proper setting of the parameters, the grid search approach is used to test all the possible combinations of parameters with different values. Bold values in Table 2 show the selected values. Since the initial instances to create the graph are selected randomly, the process of hyper-parameter tuning is repeated five times and the values with the majority in their occurrence are selected. We repeated these experiments several times first to make sure that the sub-sampled data represented the normal case and then to avoid jumping to conclusions without sufficient evidence.

5.3. Evaluation metrics

Two types of officially evaluation schemes in the contexts of *unsupervised* and *supervised* methodologies were used in SemEval2010 in order to evaluate the systems' performances. In the former approach, the induced senses are considered as clusters of instances and compared to sets of instances which have been annotated as gold senses. In this way, the v-measure (Rosenberg & Hirschberg, 2007) uses as an evaluation metric to present the harmonic mean of coverage and homogeneity of the resulted sense clusters. Homogeneity refers to the degree that each cluster consists of instances primarily belonging to a single gold sense, while completeness or coverage refers to the degree that each gold sense consists of instances primarily assigned to a single induced cluster. In addition to the v-measure as an unsupervised evaluation measure, paired f-score (Artiles, Amigó, & Gonzalo, 2009) as another evaluation metric has also been used transforming the results of clustering into the classification outputs.

The second evaluation scheme, evaluate the WSI systems' results in a WSD task. In this method, the test dataset is divide into mapping and an evaluation corpus. Mapping corpus is used to map the automatically

Table 1
Trial and Test datasets details.

	Trial			Test		
	#Target	#Instances	Average-senses	#Target	#Instances	Average-sense
Noun	3	50	7	50	5285	4.46
Verb	3	50	8.5	50	3630	3.12
Adjective	2	50	5	-	-	-

Table 2
Hyper-parameters.

	Description	Values Considered
P_1	Number of initial instances	{**25**,50,75,100}% of available set of instances of a target word
P_2	Minimum number of shared substitutes	{**3**, 4, 5, 6, 7, 8, 9}
P_3	Clustering method	{Chinese Whisper, **Leader Follower**}
P_4	Similarity measurement	{**Embedding Similarity**, Substitutes Intersection}
P_5	Similarity threshold	{**0.7**, 0.75, 0.80, 0.85, 0.9}

induced senses to the gold senses, while the other one used to evaluate the results in a WSD setting. The mapping is learned on either 80% or 60% of the test set and amounts to determining which gold sense label appears most often with an induced sense label. This process of repeated random sampling was performed to avoid the problems of providing different system rankings using different splits.

According to Manandhar and Klapaftis (2009), in the context of unsupervised evaluation, both v-measure and paired f-score show biases regarding the average number of induced clusters: v-measure is biased toward a higher number of clusters, whereas the paired f-score despite averaging between pairwise precision and recall penalizes a higher number of clusters. To account for these deficiencies, and for ease of comparison, we also report the harmonic and geometric mean of the two unsupervised measures, denoted by HAVG and GAVG, respectively. The harmonic mean can be considered as the complementary of the arithmetic mean of the reciprocals of the given set of observations while the geometric mean indicates the central tendency or typical value of a set of numbers by using the product of their values (Friehauf, Hertel, Liu, & Luong, 2013).

The SemEval2010 task organizers provided three baselines: MFS (most frequent sense - labels all test instances with the most frequent sense according to the mapping function), 1cl1inst (one cluster per instance-gives each instance its label) and Random (gives all instances a random sense label). These baselines are complementary to each other and very competitive under some of the above metrics.

6. Results

We used the official evaluation scripts presented by the organizers of SemEval2010. The first experiment encompasses the evaluation of the proposed model on full test dataset. The performance achieved by the proposed model in the context of unsupervised learning, along with the results of the highly performed state-of-the-art systems are summarized in Table 3. We also conducted other experiments trying to see the effect of each concept we used in our model as follows: graph clustering (LF) on all instances of target word making use of the intersection of substitutes for edge weighting, the proposed model for WSI (PWSI) using only the context and the same experiment with only automatically generated lexical substitutes (AutoLS). Moreover, the systems with one cluster for each instance (1cl1ins), most frequent single sense (MSF) and randomly assigned clusters for instances (Random) are also presented as baselines.

As it can be seen in Table 3 having two metrics (v-measure, f-score) makes the performance comparison process a little bit difficult. In addition, considering the average number of clusters as another metric pushes even more complexity. Thus, the harmonic and geometric mean averages of v-measure and f-score are reported as the main metrics to compare the performance of the different systems.

From the results, MFS and 1cl1inst give respectively the highest paired f-score and the v-measure, but with the other measure the results are very low. In addition, the average number of induced clusters with both baselines are unrealistic (1 and 89.15). These properties of both baselines confirm that they cannot be used as real solutions to the WSI problem. Table 3 clearly shows that doing only graph clustering on all instances of a target word, while leading to a relatively high paired f-core, but giving a very low v-measure implies its ineffectiveness. The use of PWSI with only context embedding leads to relatively better results in the cost of high number of clusters. PWSI results confirm the benefits of combining the model component instead of using them separately.

Algaic and NMF_{con} among all previously developed solutions are those that outperformed in terms of v-measure and paired f-score respectively. While these two systems are predominant based on one measurement, they do not show high performance with the other. NMF_{con} s a latent semantics-based WSI model that uses latent topical dimensions to distinguish between the different senses of a target word. Non negative matrix factorization is the method used to identify latent dimensions. NMF_{con} consisting of two different models- one for nouns and the other for verbs- for each model, they used large external resources to extract the factorization matrix. It also adopts a conservative approach to the process of selecting the candidate's senses.

However, when we compare our results with those of older studies, it is important to note that the overall harmonic and geometric averages of PWSI are significantly better than the others.

NBWSI is another system that displays promising performance in terms of HAVG/GAVG. It is a probabilistic approach involving the Bayesian models. The distinction between *NBWSI* and almost every other system, comes primarily from the use of a large number of training datasets to adjust the model's hyper-parameters and find out the probability distributions needed to make the model applicable.

Together, the current results confirm our hypothesis about the benefits of using contextual and substitutional information to improve the performance of WSI systems.

Although PWSI is the first solution ranked in terms of unsupervised evaluation, in comparison with other systems using supervised evaluation measures, we could not obtain higher results. Table 4 summarizes the supervised recall for the same state-of-the-art systems and baselines that were inspected earlier.

Given that WSI is naturally a kind of unsupervised classification task, we believe that the use of unsupervised evaluation parameters can be more reliable in order to compare system performance.

6.1. Results after post-processing

As shown in the Table 3 for all high performing systems, the average number of clusters is greater than ten, while according to the data description, the true value is approximately 3.84. To resolve this issue, we applied post-processing as explained in Section 4.3. We tried K-means with a prior number of clusters equal to 3 and 4. Table 5 and Table 6 summarize the results after post-processing with different algorithms in the context of unsupervised and supervised evaluation respectively. The results achieved by the first approach were meager and

Table 3
Performance scores of the proposed model(top section), some highly performed models (middle section), and three baselines(bottom section). Best results in each group shown in bold. The last three columns are the harmonic/geometric mean averages and the average number of induced clusters respectively. N: Noun, V: Verb, #Cl: Average number of clusters.

	v-measure			Paired f-score			HAVG	GAVG	#Cl
Model	**All**	**N**	**V**	**All**	**N**	**V**			
Only LF	.083	.101	.057	**.557**	**.505**	**.635**	.144	.215	5.45
PWSI (AutoLS)	.112	.112	.110	.514	.475	.571	.183	.239	5.57
PWSI(Context)	.170	.175	.162	.213	.218	.205	.217	.217	24.61
PWSI	**.223**	**.251**	**.181**	.411	.418	.401	**.289**	**.302**	**10.21**
Alagic (Alagić et al., 2018)	**.248**	**.312**	**.232**	.220	.214	.230	0.233	.233	13.38
NBWSI (Charniak, 2013)	.180	.237	.099	.529	.525	.535	**.268**	**.308**	3.42
HERMIT (Jurgens & Stevens, 2010)	.162	.167	.156	.267	.244	.301	.201	.207	1.87
UoY (Korkontzelos & Manandhar, 2010)	.157	.206	.086	.498	.382	.666	.238	.279	11.54
NMF_{lib} (Van de Cruys & Apidianaki, 2011)	.118	.135	.094	.453	.422	.498	.187	.231	4.80
NMF_{con} (Van de Cruys & Apidianaki, 2011)	.039	.039	.039	**.602**	**.546**	**.684**	.073	.153	**1.58**
KSUKDD (Elshamy et al., 2010)	.157	.180	.124	.369	.246	.547	.220	.240	17.5
Duluth−WSI−SVD (Pedersen, 2010)	.090	.114	.057	.411	.371	.467	.147	.192	4.15
Most frequent sense(*MFS*)	.000	.000	.000	**.635**	**.570**	**.727**	.000	.000	1.00
One cluster per instance (1*cl*1*inst*)	**.317**	**.256**	**.358**	.001	.001	.001	.001	.017	**89.15**
Random	.044	.046	.042	.319	.341	.304	**.077**	.118	4.00

Table 4
Performance scores in terms of Supervised Recall (SR) using 80%-20% setting.

	SR (80% - 20%)		
Model	All	N	V
Only LF	.593	.539	.674
PWSI (AutLS)	.596	.539	.683
PWSI (Context)	.526	.500	.566
PWSI (AutoLS + Context)	**.607**	**.552**	**.689**
Alagic (Alagić et al., 2018)	**.750**	**.707**	**.811**
NBWSI (Charniak, 2013)	.654	.626	.695
HERMIT (Jurgens & Stevens, 2010)	.583	.536	.653
UoY (Korkontzelos & Manandhar, 2010)	.624	.594	.668
NMF_{lib} (Van de Cruys & Apidianaki, 2011)	.626	.573	.702
NMF_{con} (Van de Cruys & Apidianaki, 2011)	.603	.545	.688
KSU KDD (Elshamy et al., 2010)	.521	.602	.466
Duluth−WSI−SVD (Pedersen, 2010)	.604	.689	.546
Most frequent sense (*MFS*)	**.587**	**.532**	**.666**
One cluster per instance (1*cl*1*inst*)	.000	.000	.000
Random	.573	.515	.567

Table 6
Supervised Recall scores after post-processing with different settings. AP: Affinity Propagation, km: K-means, T: Threshold, PNC: Prior Number of Clusters given to K-means.

			Supervised Recall		
CA	T	PNC	All	N	V
AP	1	-	.623	.564	.711
AP	2	-	**.624**	**.565**	**.712**
km	1	3	.621	.564	.705
km	1	4	.618	.564	.701
km	2	3	**.624**	**.564**	**.714**
km	2	4	.624	.566	.711
Base Model			**.607**	**.552**	**.689**

Table 5
Performance scores after post-processing with different settings. AP: Affinity Propagation, km: K-means, T: Threshold, PNC: Prior Number of Clusters given to K-means.

			v-measure			f-score			HAVG	GAVG	#CL
CA	**T**	**PNC**	**All**	**N**	**V**	**All**	**N**	**V**			
AP	1	-	.160	.163	.156	.414	.420	.405	.230	.257	8.25
AP	2	-	.154	.159	.147	.415	.421	.407	.224	.252	7.46
km	1	3	.159	.162	.154	.413	.419	.403	.229	.256	8.2
km	1	4	.162	.165	.157	.413	.419	.403	.232	.258	**8.6**
km	2	3	.152	.156	.146	.415	.421	.406	.222	.251	7.14
km	2	4	.157	.161	.152	.414	.421	.405	.227	.254	7.88
Base Model			**.223**	**.251**	**.181**	**.411**	**.418**	**.401**	**.289**	**.302**	10.21

not promising at all, therefor, they are discarded for being reported here. In addition, we have tried these experiments on clusters with up to two members (T shows number of instances).

Based on HAVG and GAVG in Table 5, it can be seen that using the affinity propagation for re-clustering of the induced singleton clusters reduces the average number of clusters by two while decreasing the overall performance.

Furthermore, looking at Table 6, it is also observable that the application of all sorts of post-processing may increase the overall score for supervised recalls. fastText can construct the vector of a word from its character n-grams even if the word doesn't appear in the training corpus. Instead of providing knowledge about the word types, ELMo builds a context-dependent, and therefore instance-specific embedding. Tables 7 and 8 and demonstrate the results on using fastText [8] and ELMo[9] instead of word2vec.

As Tables 7 illustrates, making use of both fastText and ELMo instead of word2vec outperform our best model as expected. The approaches of fastText and word2vec in word representation are substantially similar

[8] We made use of pretrained fastText with 1 million word vectors trained with sub word information on Wikipedia 2017, UMBC web base corpus and statmt.org news dataset downloaded from: https://fasttext.cc/docs/en/english-vectors.html

[9] The medium-sized ELMo trained on approximately 800 million WMT 2011 crawl data chips is used. https://allennlp.org/elmo

Table 7
Performance scores of the proposed model replacing ELMo and fastText with word2vec.

	v-measure			Paired f-score			HAVG	GAVG	#Cl
Model	**All**	**N**	**V**	**All**	**N**	**V**			
PWSI(fastText)	.231	.263	.192	.434	.456	.409	.301	.316	9.4
PWSI(Elmo)	**.224**	**.271**	**.204**	**.469**	**.489**	**.442**	**.322**	**.339**	**8.7**
PWSI	.223	.251	.181	.411	.418	.401	.289	.302	10.21

Table 8
Performance scores in terms of Supervised Recall making use of fastText and ELMo.

	SR		
Model	**All**	**N**	**V**
PWSI(fastText)	.624	.561	.721
PWSI(ELMo)	**. 641**	**.574**	**.743**
PWSI	.607	.552	.689

Table 9
Results of different systems on the target word *swear*.

	# Sense groups	v-measure
1cl1inst	44	.575
MFS	1	0.00
Random	4	.137
Only LF	13	.237
PWSI(ELMo)	**9**	**.312**

together, therefore the improvement obtained by using fastText when comparing to ELMo is low. The lower average number of clusters, as well as the improved v-measure and paired f-score confirm the essential role of the word embedding model within the proposed model. Considering the improvements by replacing ELMo as a context sensitive word embedding model with word2vec in the context of supervised and unsupervised evaluations may be further evidence confirming our assumptions as to the importance of contextual information for WSI.

As an example, the results of different systems on a target word *swear* from test set are summarized in Table 9. It has 44 instances with the sentences' lengths in the range of 6 to 28 with 5 gold senses altogether. As it can be seen in Table 9, regardless of considering MFS that assigns all the instances to the only one sense group, applying the PWSI(ELMo) results in closets number of the clusters to the number of gold senses after random model. In addition it leads to the highest v-measure among the others.

Analysis of outcomes at the instance level shows that, for most singleton clusters, their member instances are very short and sometimes in an unusual format.

For example, *Vince swore.*, [*Chuck*$_S$*chummer* :] *Weigh in.*, and *had it on the air*. For the target words *Swore*, *weigh* and *air* respectively. The problem of finding word-embedding vectors in such short cases is one of the reasons for miss-clustering. On the other hand, combining these singleton clusters with their neighbors in addition to reducing the number of clusters results in an increase in supervised recall evaluation.

7. Conclusion

Making use of lexical substitutions along with the context of a target word showed promising results in word sense induction problem. In this article, we have presented a novel approach to word sense induction combining the graph clustering - to find the primary sense groups - and instance-sense assignment based on the cosine similarity between the embedding vectors of instances and basic sense groups' representatives. It is also possible to create new sense groups during the latter process if the similarity measures can not satisfy the predefined threshold. For the graph clustering step, we made use of Leader–Follower algorithms borrowed from the community detection domain with no need to have prior information about the number of sense clusters.

Our experiments on the SemEval2010 dataset show that our model leads to state-of-the-art results in terms of unsupervised evaluation measures with a less average number of induced clusters. The simplicity and applicability of applying the proposed model on newly coming instances of a target word with no need to recompute everything from scratch can be considered as advantages of our model. The recent property of our model can be considered as an on-line word sense induction. Our model is essentially based on the quality of automatically generated lexical substitutes, so the more appropriate the substitutes generated, the higher the performance of the WSI. Most of the miss-induced senses of target words by our model were those inside very short, incomplete, and unusual contexts. Making use of some hand crafted linguistics features beside the word and contextual embedding vectors can be suggested as the next step to improve the results. In addition, taking advantage of fine-tuned context-sensitive word representation models like BERT, ELMo, etc. along with improved automatic lexical substitution systems can also outperform the WSI models.

CRediT authorship contribution statement

Abbas Akkasi: Conceptualization, Methodology, Writing - review & editing. **Jan Snajder:** Data curation, Supervision, Validation.

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

This work has been fully supported by the Croatian Science Foundation under the project UIP- 2014–09-7312.

References

Alagić, D., Šnajder, J., & Padó, S. (2018). Leveraging lexical substitutes for unsupervised word sense induction. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Amrami, A., & Goldberg, Y. (2019). Towards better substitution-based word sense induction. arXiv preprint arXiv:1905.12598.

Artiles, J., Amigó, E., & Gonzalo, J. (2009). The role of named entities in web people search. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2 (pp. 534–542). Association for Computational Linguistics.

Bartunov, S., Kondrashkin, D., Osokin, A., & Vetrov, D. (2016). Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics* (pp. 130–138).

Başkaya, O., & Jurgens, D. (2016). Semi-supervised learning with induced word senses for state of the art word sense disambiguation. *Journal of Artificial Intelligence Research, 55*, 1025–1058.

Baskaya, O., Sert, E., Cirik, V., & Yuret, D. (2013). 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) (pp. 300–306). volume 2.

Biemann, C. (2006). Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In Proceedings of the first

workshop on graph based methods for natural language processing (pp. 73–80). Association for Computational Linguistics.
Biemann, C., & Nygaard, V. (2010). Crowdsourcing wordnet. In The 5th International Conference of the Global WordNet Association (GWC-2010).
Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.
Brody, S., & Lapata, M. (2009). Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 103–111). Association for Computational Linguistics.
Charniak, E., et al. (2013). Naive bayes word sense induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1433–1437).
Cocos, A., & Callison-Burch, C. (2016). Clustering paraphrases by word sense. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1463–1472).
Van de Cruys, T., & Apidianaki, M. (2011). Latent semantic word sense induction and disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 1476–1485). Association for Computational Linguistics.
Das, P., Das, A. K., Nayak, J., Pelusi, D., & Ding, W. (2019). A graph based clustering approach for relation extraction from crime data. *IEEE Access, 7*, 101269–101282.
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.
Di Marco, A., & Navigli, R. (2013). Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics, 39*, 709–754.
Dorow, B., & Widdows, D. (2003). Discovering corpus-specific word senses. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2* (pp. 79–82). Association for Computational Linguistics.
Dueck, D. (2009). Affinity propagation: clustering data by passing messages. *Citeseer*.
Elshamy, W., Caragea, D., & Hsu, W. H. (2010). Ksu kdd: Word sense induction by clustering in topic space. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 367–370). Association for Computational Linguistics.
Erk, K., McCarthy, D., & Gaylord, N. (2009). Investigations on word senses and word usages. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1 (pp. 10–18). Association for Computational Linguistics.
Ferraresi, A., Zanchetta, E., Baroni, M., & Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google (pp. 47–54).
Firth, J. R. (1935). The technique of semantics. *Transactions of the Philological Society, 34*, 36–73.
Friehauf, M., Hertel, M., Liu, J., & Luong, S. (2013). On compass and straightedge constructions: Means.
Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks, 18*, 602–610.
Harris, Z. S. (1954). *Distributional structure. Word, 10*, 146–162.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*, 1735–1780.
Hope, D., & Keller, B. (2013). Maxmax: a graph-based soft clustering algorithm applied to word sense induction. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 368–381). Springer.
Hope, D., & Keller, B. (2013). 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) (pp. 689–694). volume 2.
Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes: the 90% solution. In Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers (pp. 57–60). Association for Computational Linguistics.
Jurgens, D., & Klapaftis, I. (2013). Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (pp. 290–299). volume 2.
Jurgens, D., & Stevens, K. (2010). Hermit: Flexible clustering for the semeval-2 wsi task. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 359–362). Association for Computational Linguistics.
Klapaftis, I. P., & Manandhar, S. (2008). Word sense induction using graphs of collocations. In *ECAI* (pp. 298–302).
Klapaftis, I. P., & Manandhar, S. (2010). Word sense induction & disambiguation using hierarchical random graphs. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 745–755). Association for Computational Linguistics.
Korkontzelos, I., & Manandhar, S. (2010). Uoy: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 355–358). Association for Computational Linguistics.
Lau, J. H., Cook, P., McCarthy, D., Newman, D., & Baldwin, T. (2012). Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 591–601). Association for Computational Linguistics.
Manandhar, S., & Klapaftis, I. P. (2009). Semeval-2010 task 14: Evaluation setting for word sense induction & disambiguation systems. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions* (pp. 117–122). Association for Computational Linguistics.
Manandhar, S., Klapaftis, I. P., Dligach, D., & Pradhan, S. S. (2010). Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 63–68). Association for Computational Linguistics.
McCarthy, D., & Navigli, R. (2007). Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 48–53). Association for Computational Linguistics.
McCarthy, D., & Navigli, R. (2009). The english lexical substitution task. *Language Resources and Evaluation, 43*, 139–159.
Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 51–61).
Melamud, O., Levy, O., & Dagan, I. (2015). A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (pp. 1–7).
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM, 38*, 39–41.
Miller, G. A. (1998). *WordNet: An electronic lexical database*. MIT press.
Mills, M. T., & Bourbakis, N. G. (2013). Graph-based methods for natural language processing and understanding a survey and analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44*, 59–71.
Mitra, S., Mitra, R., Riedl, M., Biemann, C., Mukherjee, A., & Goyal, P. (2014). That's sick dude!: Automatic identification of word sense change across different timescales. arXiv preprint arXiv:1405.4392.
Nasiruddin, M. (2013). A state of the art of word sense induction: A way towards word sense disambiguation for under-resourced languages. arXiv preprint arXiv: 1310.1425.
Nasiruddin, M., Schwab, D., Tchechmedjiev, A., Sérasset, G., & Blanchon, H. (2014). Induction de sens pour enrichir des ressources lexicales. In 21ème conférence sur le Traitement Automatique des Langues Naturelles (TALN 2014) (p. 6).
Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR), 41*, 10.
Navigli, R., & Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 116–126). Association for Computational Linguistics.
Navigli, R., & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*, 678–692.
Ngomo, A.-C. N., & Schumacher, F. (2009). Borderflow: A local graph clustering algorithm for natural language processing. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 547–558). Springer.
Pedersen, T. (2007). Umnd2: Senseclusters applied to the sense induction task of senseval-4. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 394–397). Association for Computational Linguistics.
Pedersen, T. (2010). Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 363–366). Association for Computational Linguistics.
Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.
Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018b). Deep contextualized word representations. In *Proceedings of NAACL-HLT* (pp. 2227–2237).
Ramprasad, S., & Maddox, J. (2019). Coke: Word sense induction using contextualized knowledge embeddings. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.
Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.
Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics, 24*, 97–123.
Shah, D., & Zaman, T. (2010). Community detection in networks: The leader-follower algorithm. arXiv preprint arXiv:1011.0774.
Takano, Y., Iijima, Y., Kobayashi, K., Sakuta, H., Sakaji, H., Kohana, M., & Kobayashi, A. (2019). Improving document similarity calculation using cosine-similarity graphs. In *International Conference on Advanced Information Networking and Applications* (pp. 512–522). Springer.
Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141–188.
Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language, 18*, 223–252.
Widdows, D., & Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1–7). Association for Computational Linguistics.
Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks, 16*, 645–678.
Zhang, M. (2014). In *Word sense induction for machine translation In Proceedings of the 28th Pacific Asia Conference on Language*. Information and Computing.