

# Using a Heterogeneous Linguistic Network for Word Sense Induction and Disambiguation

Edmundo-Pavel Soriano-Morales, Julien Ah-Pine, Sabine Loudcher

Univ Lyon  
Université Lumière Lyon 2  
ERIC EA 3083  
F-69676, France

{edmundo.soriano-morales, julien.ah-pine, sabine.loudcher}@univ-lyon2.fr

**Abstract.** Linguistic Networks are structures that allow us to model the characteristics of human language through a graph-like schema. This kind of modelization has proven to be useful while dealing with natural language processing tasks. In this paper, we first present and discuss the state of the art of recent semantic relatedness methods from a network-centric point of view. That is, we are interested in the types of networks used to solve practical semantic tasks. In order to address some of the short-comings in the studied approaches, we propose a hybrid linguistic structure that takes into account lexical and syntactical language information. We show our model’s practicality with a proof of concept: we set to solve word sense disambiguation and induction while using the presented network schema. Our modelization aims to shed light into ways of combining distinct types of linguistic information in order to take advantage of each of its components’ unique characteristics.

## 1 Introduction

Today, thanks to the global pervasiveness of the Web, we have access to large quantities of open and collaborative sources of textual data. Nonetheless, all this information, in the form of language units and related linguistic attributes, requires a proper method for representation, querying and analysis while taking into account the attribute of each unit within its local and global context. With these requirements in mind, researchers have used language networks<sup>1</sup> since a long time [1] to model linguistic complex data. Indeed, a complex network allows us to have a look at the information from both local and global perspectives. However, it is not until recently, with the growth of computational power, that we are able to exploit linguistic networks at a larger level. In short, in order to extract useful knowledge from a linguistic network we need a representation that can combine diverse kinds of language attributes (and the relations among them), as well as facilitate the application of graph analytic algorithms.

---

<sup>1</sup> In general, we use the term *network* and *graph* interchangeably. However, in some cases we consider a network as being represented by a graph structure, among other properties.

The characteristics of a linguistic network varies according to the necessities of the natural language processing (NLP) task we are trying to solve. Still, we can identify two general aspects: the type of network used to hold the information and the algorithms applied on it to extract new insights. In this context, the work presented in this article has two goals: (1) review recent linguistic network models used to resolve semantic NLP tasks, and (2), propose a novel linguistic network that addresses some of the structural limitations of the works studied.

Accordingly, we first provide a simplified and organized state of the art of linguistic networks in the domain of Word Sense Disambiguation and Induction (WSD and WSI). As we will find out, relatively few approaches go beyond using classic lexical co-occurrence information as a source to discriminate contexts and different senses of a word. It is our intuition that by leveraging different types of linguistic relations we can obtain more pertinent results on a given semantic task. In this sense, we propose a network model that is able to hold diverse kinds of language information and allow for a simple manipulation of the data contained in it. Using this schema, we perform a proof of concept to illustrate the advantages of using such structures for the word sense disambiguation and word sense induction tasks. In that respect, it is not our goal to compete against the best systems for WSD and WSI described in the literature and which require tuning parameters most of the time. Our aim is rather to show that using linguistic networks enables us to encode more fine-grained language information that we could leverage to better address NLP tasks in comparison to basic lexical co-occurrences information.

We organize the paper as follows, in Section 2 we introduce basic concepts and in Section 3 we review network-based approaches to semantic-similarity tasks specifically from a graph-centric view. In Section 4 we propose a linguistic network based on hypergraphs. Next, we show the potential utility of such network in Section 5. Finally, we present our conclusion and future research in Section 6.

## 2 Background

Below we will delineate the preliminary concepts used throughout the rest of our paper. We introduce the concept of linguistic network (or language network) as well as the semantic tasks we are interested in.

*Linguistic Network* We define a Linguistic Network (LN) as a modelization of the human language in terms of a graph structure. Usually, textual entities (e.g., letters, words, phrases) are linked together by means of grammatical or semantic relations [2]. A network structure allows us to study the characteristics of said relations in order to extract useful knowledge from them.

In this work we focus on two aspects of a linguistic network: the type of LN, with regards to its contents, as well as the graph algorithms used over the network to solve a given NLP task. In this work we concentrate on word-semantics related tasks. Among these tasks, two that are highly popular are word sense disambiguation and word sense induction and.

*Word Sense Disambiguation (WSD)* Given a target word  $tw$ , a context  $ct$ , and a set  $mn$  containing possible meanings for  $tw$ , the goal of WSD is to determine which signification corresponds to  $tw$  from the set  $mn$  according to the context  $ct$ . This task is usually solved leveraging a dictionary or thesaurus that establishes semantic links between word senses. This type of resource is also known as Lexical Knowledge Base (LKB)<sup>2</sup>. A LKB can be defined as an ontology that relates words according to their semantic relation. Two quintessential examples of a LKB are the Wordnet semantic dictionary [3] and BabelNet [4].

*Word Sense Induction (WSI)* The methods employed to solve WSD are generally unsupervised, that is, they do not require an annotated corpus to infer the appropriate sense for a given word. Nonetheless, a certain level of supervision can be distinguished on these approaches. Indeed, LKBs are, most of the time, built using human supervision. In order to circumvent this constraint, researchers have devised fully unsupervised techniques to automatically find the senses  $mn$  of a word  $tw$  by leveraging a background corpus. Once the senses have been induced, these approaches perform WSD. This task is named Word Sense Induction (WSI).

### 3 State of the Art

According to their objectives, we can consider two types of contributions in the linguistic-network literature [2]: on the one hand, there are those approaches that investigate the nature of language via a graph representation, and on the other hand, we find those that propose a practical solution to a given NLP problem. In that regard we can cite the following survey papers [5,6,7,8].

This article focuses on the latter type of approaches. Moreover, we pay particular attention to two aspects of a given network-based technique: (1) the characteristics of the linguistic data within the network, and (2), the algorithms used to extract knowledge from it.

Once the LN modelization concept and the concerned tasks are introduced, we move on to the content of our literature review. As we defined before, a LN comprises two main characteristics: the type of language network and the nature of the algorithms used in each network.

#### 3.1 Types of Linguistic Networks

In the following paragraphs we introduce the general categories of LNs according to their type of content and relations. We will introduce these categories as well as the approaches that make use of them.

In [8] they define four types of LNs: co-occurrence network, dependencies network, semantic network and similarity network. Meanwhile, from a deeper linguistic point of view, [2] defines broader categories, each having several sub-types.

---

<sup>2</sup> We note that in our context, a LKB has the same characteristics of a semantic linguistic network. Thus, we employ both terms interchangeably.

The main difference (in our context) between both definitions lies in the separation of categories. In [2], they conflate syntactic-dependency and co-occurrence networks into the same category: word co-occurrence networks. Similarly, they join semantic and similarity networks together and place them inside a broader category of lexical networks. The third family defined concerns phonological networks which is out of the scope of this paper. In this work we will explore five categories of linguistic networks: semantic, lexical co-occurrence, syntactic co-occurrence and heterogeneous networks. The following sections will elucidate what each kind of network represent, we will mention works that employ this kind of networks and also list the main methodology differences that variate from one approach to another.

*Semantic Networks* A Semantic Network (SN) relates words, or concepts, according to their meaning. The classical example of a SN is the renowned knowledge base Wordnet. This network, which serves also as an ontology, contains sets of synonyms (called synsets) as vertices and semantic relations as their edges. Typical semantic relationships include synonym-antonym, hypernym-hyponym, holonym-meronym. However, other semantic similarities can be defined. The edges are usually not weighted, although in some cases certain graph similarity measures may be used.

Word sense disambiguation is indeed a task usually solved using semantic networks, specially Wordnet (and to lesser extent, BabelNet) [9,10,11,12,13] Given an input text with a set of ambiguous target words to process, these approaches follow a two-step algorithm:

1. Link target words (usually nouns, without stop-words and functional words) with their corresponding sense (or synset in the case of Wordnet-like dictionaries) and extract their vertices and edges into a new, smaller, SN.
2. Apply a node ranking technique, usually a random walk based method, and select, for each ambiguous word in the input text, its top ranking synset node as the correct sense.

*Lexical Co-occurrence Networks* Most co-occurrence based intuitions in NLP have their origin in the distributional hypothesis [14]. The idea is resumed by the well know phrase "a word is characterized by the company it keeps" [15]. That is to say, words with similar neighbor words (or contexts) tend to be semantically similar.

This intuition has been exploited deeply in NLP. One of the most effective ways of representing word co-occurrences is by means of a graph structure. Indeed, this kind of graphs are the central column of a Lexical Co-occurrence Network (LCN). In these structures, nodes represent words and edges indicate co-occurrence between them, i.e., two words appear together in the same context. A context can vary from a couple of words (before or after a given word) to a full document, although it is usually defined at sentence level. The edges' weight represent the strength of a link and is generally a frequency based metric that takes into account the number of apparitions of each word independently and together.

To solve a task in a completely unsupervised way, researchers generally use this kind of networks instead of LKBs. It is then natural that word sense disambiguation approaches leverage lexical co-occurrence networks, and in return, the distributional hypothesis, to automatically discover senses for a given target word. That is why WSI methods [16,17,18] are tightly related to LCNs. The cited works use a LCN as described before while other works such as [12,19] represent the co-occurrence by means of a hypergraph schema. In short, a hypergraph structure is a graph generalization where an edge (called hyperedge) can link multiple vertices per edge and thus it is able to provide a more complete description of the interactions between several nodes [20].

WSI systems generally perform four steps. Given an input text with a set of target words and their contexts (target words must have several instances throughout the document to cluster them), the steps are the following:

1. Build a LCN, assigning tokens as nodes and establishing edges between them if they co-occur in a given context (usually if they both appear in the same sentence).
2. Determine the weights for each edge according to a frequency metric.
3. Apply a graph clustering algorithm. Each cluster found will represent a sense of the polysemous word.
4. Match target word instances with the clusters found by leveraging each target word context. Specifically, assign a cluster (a sense) to each instance by looking at the tokens in the context.

*Syntactic Co-occurrence Networks* A Syntactic Co-occurrence Network (SCN) is very similar to a LCN in the sense that both exploit the distributional hypothesis. Nonetheless, SCNs go further by leveraging syntactic information extracted from the text. There are two main types of syntactic information both represented as tree structures: constituency-based parse trees and dependency-based parse trees. Briefly, the former structure splits a phrase into several sub-phrases. In this way we can get a glimpse of the role of each word inside a phrase. The latter tells us about the relationships existing between words in the phrase. SCNs employ, most of the time, dependency trees to create a graph that relates words according to their syntactic relations. In the case of [21], a graph is built using syntactic dependencies. It is used to perform WSI using a very similar approach as those systems using LCNs. We note that approaches based on SCNs are scarcely used in WSD or WSI systems, and therefore they are an interesting research avenue to explore.

## 4 Heterogeneous Linguistic Network: Our Proposal

In the previous section we have mentioned two disadvantages found in the language networks covered in Section 3. Namely, the lack of syntactic information and the homogeneous nature of the networks. In this section we propose a language network that, at this point of our research, addresses both of these concerns. Building upon previous linguistic representations [17,22,19], our model

is based on the use of a hypergraph. Hypergraphs have been employed in the literature to model complex systems. Their single most important difference, being able to relate more than two vertices at the same type, allows for a better characterization of interactions within a set of individual elements (in our case, words) [23].

Indeed, our hypergraph modelization integrates four types of relations between tokens: sentence co-occurrence, part-of-speech tags, words’ constituents data and dependency relations in a single linguistic structure. We group words together according to the these features.

Formally, a hypergraph is a generalization of a graph defined as a tuple  $G = (V, E)$ , where the vertices  $V = \{v_1, v_2, \dots, v_n\}$  represent the set of nodes and  $E = \{e_1, e_2, \dots, e_m\}$  the set of hyperedges which contain links between one or more vertices [24].

In our case, the set of tokens in the corpus are the set of nodes  $V$ , and the set of hyperedges  $E$  represent the relations between nodes according to different linguistic aspects. Each hyperedge may be one of three types: noun phrase<sup>3</sup> constituents (*CONST*), dependency relations (*DEP*), or sentence context (*SEN*). We consider that a token  $v$  belongs to a hyperedge of type NP or SEN if the token appears in the same noun phrase or in the same sentence. A token  $v$  belongs to a hyperedge of type DEP if it is the dependent of a certain dependency relation coupled with its corresponding head (or governor). The hypergraph can be represented as a  $n \times m$  incidence  $H$  matrix with entries  $h(i, j) = N(v_i, e_j)$  where  $N(v_i, e_j)$  is the number of times  $v_i \in e_j$  occurs in the corpus.

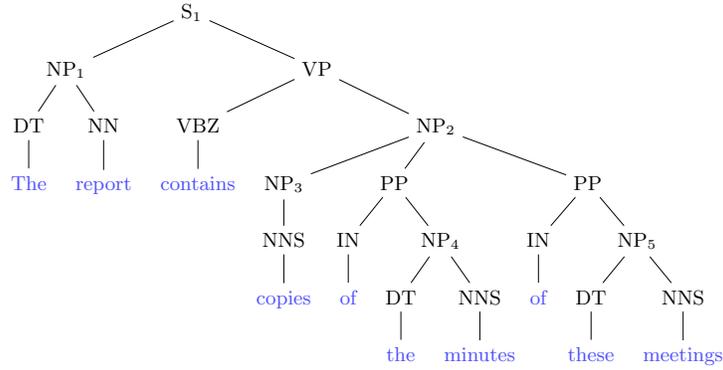
We illustrate our hypergraph incidence matrix with the following example phrase: *The report contains copies of the minutes of these meetings..* We tokenize the phrase, keeping all the words, and we lemmatize and parse it to obtain both constituency and dependency trees.

The constituency tree of the example phrase is shown in Figure 1. The sentence, as well as each noun phrase (NP) node is identified by a number. We can observe that this phrase is composed by five noun phrases (NP) and one verb phrase. Meanwhile, some of the NPs are formed by other kind of phrases, depending on the grammar production rule used to build each one of them. As is usual in this kind of structures, there is a one to one relation between the number of tokens in the sentence and the number of leaves in the tree.

The dependencies of the example phrase are shown in Table 1. They indicate the syntactic relation between the governor of a phrase and a dependent. In these relations’ examples, the head is the first token to appear followed by the dependent word.

From both of these types of information we can build a hypergraph representation as stated before. The incidence matrix is illustrated in Table 2. For brevity, we only show nouns as well as only the first three noun phrases and the nominal subject (*nsubj*) and direct object (*dobj*) dependency relations. Looking at the table, we can therefore infer that the word *copies* appears in two hyper-

<sup>3</sup> In this work we consider only noun phrases (NPs). Still, we can easily add other type of phrase chunks.



**Fig. 1.** Constituency-based tree of the phrase *The report contains copies of the minutes of these meetings*.

**Table 1.** Dependency relations of the example phrase.

|                                 |                                 |
|---------------------------------|---------------------------------|
| <b>root</b> (root, contains)    | <b>det</b> (minutes, the)       |
| <b>det</b> (report, The)        | <b>nmod</b> (copies, minutes)   |
| <b>nsubj</b> (contains, report) | <b>case</b> (meetings, of)      |
| <b>dobj</b> (contains, copies)  | <b>det</b> (meetings, these)    |
| <b>case</b> (minutes, of)       | <b>nmod</b> (minutes, meetings) |

edges of type CONST: NP<sub>2</sub>, which is built from a NP, and two prepositional phrases (PP). Also, we see that it is part of NP<sub>3</sub>, which indicates a plural noun (NNS). Regarding the syntactic dependency hyperedges, the word *copies* appear in the *dobj\_contains* column which indicates the *copies* was indeed the direct object of the verb *contains*, Finally, we can know that *copies* appeared in the same sentence S<sub>1</sub> as the other four noun words.

**Table 2.** Incidence matrix of the example phrase hypergraph modelization.

|                  | CONST           |                 |                 | DEP      |          | SEN            |
|------------------|-----------------|-----------------|-----------------|----------|----------|----------------|
|                  | NP <sub>1</sub> | NP <sub>2</sub> | NP <sub>3</sub> | nsubj    | dobj     | S <sub>1</sub> |
|                  | DT:NN NP:PP:PP  |                 |                 | contains | contains |                |
| report           | 1               |                 |                 | 1        |          | 1              |
| contains         |                 |                 |                 |          |          | 1              |
| <b>NN</b> copies |                 | 1               | 1               |          | 1        | 1              |
| minutes          |                 | 1               |                 |          |          | 1              |
| meetings         |                 | 1               |                 |          |          | 1              |

## 5 Proof of concept: Word Sense Induction and Disambiguation

In this section we carry out a proof of concept experiment to verify the potential of our proposed network modelization. We use the task of word sense induction and disambiguation as an application context for our procedure. As stated before, we do not aim to create a system able to beat the reviewed WSD or WSI techniques. Instead, our goal is to show that using other kinds of language information we can improve the results of those obtained while using classic lexical co-occurrence, and thus emphasize the utility of using diverse linguistic information, in our case through a language hypergraph structure.

### 5.1 Methodology

The task is the following: we are given a document  $d$  with several target words  $tw$  and multiple paragraph instances for each  $tw$ . We consider each of these paragraphs as the context  $ct$  of a target word  $tw$ . The goal is to first automatically determine a set of senses for a given  $tw$  (WSI), and then assign one meaning to each of its instances (WSD).

As described before, WSI (including WSD) is usually solved following four steps: (1) creation of a linguistic network, (2) determine the level of similarity between nodes within the network, (3) cluster nodes together, thus creating individual senses, and (4) assign a cluster (sense) to each instance of a target word in the input document.

In our process, we follow a similar approach to those used in [16,17]. In short, these methods build a network of lexical co-occurrence with a background corpus and then exploit the real-world characteristics of said networks by theorizing that there are certain important nodes (called hubs) that carry a significant role among the words contained in the network and therefore may represent, coupled with their neighbors, a sense for a given target word.

In our approach, we generate a network for each  $tw$  and the high-degree nodes found inside this network ideally represent a  $tw$  sense. As presented in the previous sections, we use a hypergraph structure, similar to the one used in [17]

*Creation of the linguistic network* In the previous sections we worked with the English Wikipedia as background corpus to build and model our proposed linguistic network. Given the large size of Wikipedia, and to iterate faster our experiments, we decided to change the corpus to one with a more manageable size. We use the Open American National Corpus (OANC) [25] as background document collection to build a hypergraph network  $G_H$  following our proposed model. The OANC includes texts from several domains and encompasses 11,406,155 words. We split the documents in the corpus in sentences, then we tokenize and parse them with Stanford’s CoreNLP [26]. As described before, the dependency and constituency tree are used to build the hypergraph: words are depicted by nodes, and they may exist inside any of the three different types of hyperedges defined: sentence, noun phrase or dependency contexts. If any hyperedge is repeated

through the corpus, we increment a counter and keep the number of apparitions instead of adding redundant columns to the hypergraph incidence matrix.

At each step, that is, for each  $tw$  in the input document, we extract a sub-graph  $G_{tw}$  from  $G_H$  that contains all the words that appear together with  $tw$  (line 2), whether by lexical or syntactic co-occurrence. The  $tw$  is removed from  $G_{tw}$ . In this approach we focus specifically on dependency relations and lexical co-occurrence.

*Computing similarity between nodes* In order to computationally treat  $G_{tw}$ , we first induce a bipartite graph  $B_{tw} = (U, W, E)$  from  $G_{tw}$  (line 3). The set of left nodes  $U$  represent words and the set of right nodes  $W$  depicts the membership to a given hyperedge. Thus, we have as many nodes in  $W$  as we had hyperedges in  $G_H$ .

We compute the Jaccard index between each node  $n_i, n_j \in U$  as  $Jaccard(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$ ,  $N(j)$  being the neighbors of  $n_j$ . We use this metric in order to build a  $|U| \times |U|$  similarity matrix  $S_{tw}$  (line 4). We induce from  $S_{tw}$  a new filtered hypergraph incidence matrix  $F_{tw}$  (line 5), which contains word nodes as rows and columns as hyperedges. Each of these hyperedges represent a set of words that are deemed similar between them according to their Jaccard index value, which must be equal or higher than an assigned threshold  $th_1$ .

*Clustering words together* Once the incidence matrix  $F_{tw}$  is built we can proceed to induce senses for a target word by clustering words (vertices) together. First, we calculate the degree of each node  $n_i \in F_{tw}$ . The degree of a node is simply the number of hyperedges it is incident in. Nodes are sorted in descending order and evaluated one by one. Each node is considered as a candidate sense hub (line 6). We accept or reject a node  $n \in F_{tw}$  as a sense carrying word according to two thresholds:  $th_2$  and  $th_3$ . The former (line 9) is the minimum degree a node must have, which is automatically determined by taking into account a node if its degree is superior to the 85th percentile among all the calculated degrees. The latter (from line 11 to 17) sets a minimum limit to the average of the Jaccard similarities between each pair of neighbors of node  $n \in F_{tw}$ , within each hyperedge  $n$  belongs to. Formally, for a node  $n$ , we define the average Jaccard measure as:

$$AvgJaccard(n) = \frac{1}{|hedges(n)|} \sum_{h \in hedges(n)} \frac{\sum_{\substack{i \in h \\ j \in h; i \neq j}} Jaccard(i, j)}{|h|}$$

where  $hedges(n)$  is the set of hyperedges  $n$  is incident in and its cardinality is defined as  $|hedges(n)|$ .  $|h|$  is the number of nodes in hyperedge  $h$ .

If node  $n$  satisfies both thresholds  $th_2$  and  $th_3$ , it is deemed as a sense purveyor and all its neighbors (words that appear in the same hyperedges as  $n$ ) are conflated into a single set representing a  $tw$  sense. This new sense is added to  $SoS_{tw}$  (line 17). The sense set is then removed from  $F_{tw}$ .

The process is repeated until no more nodes satisfy both boundaries. When the process is complete, we obtain a set of senses  $SoS_{tw}$  where each set contains words that ideally represent a unique meaning for each target word.

*Sense assignment* The assignment of a sense consists in looking at each  $tw$  instance represented by a context  $ct$  and simply determining which sense  $s$  in  $SoS_{tw}$  shares the highest amount of words with  $ct$ . The sense  $s$  is thus assigned to that instance. If two senses in  $SoS_{tw}$  share the same amount of words with  $ct$ , one of them is randomly chosen. This operation is repeated for each instance of each target word.

---

**Algorithm 1:** Pseudo-code of our WSD/WSI network-based approach

---

**Input:** A set  $tw\_set = \{tw_1, tw_2, \dots, tw_n\}$  of target words. A background linguistic network  $G_H$ . Filtering thresholds  $th_1, th_2, th_3$

**Output:** A set  $SoS_{tw}$  of senses for each target word

```

1 foreach target word  $tw$  in  $tw\_set$  do
2    $G_{tw} = \text{extract\_subgraph}(G_H, tw)$ ;
3    $B_{tw} = \text{induce\_bipartite\_graph}(G_{tw})$ ;
4    $S_{tw} = \text{sim\_matrix}(B_{tw})$ ;
5    $F_{tw} = \text{induce\_hypergraph}(S_{tw}, th1)$ ;
6    $candidate\_hubs = \text{sort}(\text{degree}(F_{tw}))$ ;
7    $SoS_{tw} = []$ ;
8   foreach candidate\_hub in candidate\_hubs do
9     if  $\text{degree}(candidate\_hub) < th2$  then
10      | continue;
11       $candidate\_hyperedges = \text{get\_hyperedges}(candidate\_hub, F_{tw})$ ;
12       $candidate\_avg\_jaccard = 0$ ;
13      foreach hyperedge in candidate\_hyperedges do
14        |  $candidate\_avg\_jaccard += \text{get\_avg\_jaccard}(hyperedge)$ ;
15      end
16      if  $candidate\_jaccard > th3$  then
17        |  $SoS_{tw}.add(\text{get\_words}(candidate\_hyperedges))$ ;
18        |  $F_{tw} = F_{tw} \setminus candidate\_hyperedges$ ;
19      end
20    return  $SoS_{tw}$ 
21 end

```

---

## 5.2 Experiments and Results

The objective of this proof of concept is to show the advantages of using syntactic co-occurrence information compared to simple lexical co-occurrence. To this end, we solve the word sense induction and disambiguation tasks using the method described in the previous subsection. We create two independent systems: LEX, which uses lexical co-occurrence hyperedges, and DEP, which employs syntactic dependency hyperedges.

As evaluation dataset, we employ the data provided for Task 02 of Semeval-2007 [27] which evaluated word sense induction systems. The data consists on 100 target words<sup>4</sup> (65 verbs and 35 nouns), each target word having a set of paragraph contexts where it appears. From the available performance assessing techniques, supervised and unsupervised, we are interested in the unsupervised evaluation, which is rated using the F-score produced by an evaluation script. We also modify it to obtain also the precision and recall measures to build a precision-recall curve.

Each type of language information has its own characteristics. The sub-network formed by sentence hyperedges tends to have a much smaller number of nodes (words) than those of the dependency type. This make sense as sentences usually contain a few words, meanwhile a dependency hyperedge may incorporate upwards to hundreds of words that are related to a word by the same dependency relation. These characteristics affect the similarity between vertices and thus drove us to set the threshold ( $th1$  and  $th2$ ) values for LEX and for DEP in function of the percentile of the node’s degree and similarity values distributions, respectively.

This leaves only one threshold left,  $th_3$ . We experiment with two different ranges of values, one for each system. For DEP we set the range  $[0.3, 0.65]$  with a step of 0.05. For LEX we set  $[0.01, 0.08]$  with a step of 0.01. These ranges were chosen experimentally with two constraints in mind: (1) lower threshold values usually gave the same results<sup>5</sup> as those already included in our ranges, and (2) higher threshold values forced the system to either give only one sense per word (resulting in the most frequent baseline), or even worse, not accepting any sense, thus having a null solution. Again, we evaluate our systems by means of the (unsupervised) F-score and a precision-recall curve which provides a deeper analysis of the performance of each system while considering the variation of threshold  $th_3$ .

The F-score of both systems, and the average number of clusters (senses) produced, is shown in Figure 2. Indeed, in our experiment, the dependency based model DEP preformed better than LEX using classic lexical co-occurrence. We include the result of the UOY system [17] as a similar-method benchmark. In UOY, two background corpora are also used to build a linguistic network of lexical co-occurrences. One of the corpus is the same as the one we used to evaluate. This allows their system to induce the exact senses used in each target word instance. While this is a practical idea, we, by using a large, multiple-domain corpus, are able to induce word senses that may not even be used in the Semeval dataset. Concerning the thresholds, we use percentiles to automatically adapt to the characteristics of the hyperedges, as the lexical and dependency co-occurrences hyperedges behave differently within the linguistic network.

In Figure 3 we appreciate that even while using different threshold values, we achieve, in general, better recall and precision by using syntactic dependencies.

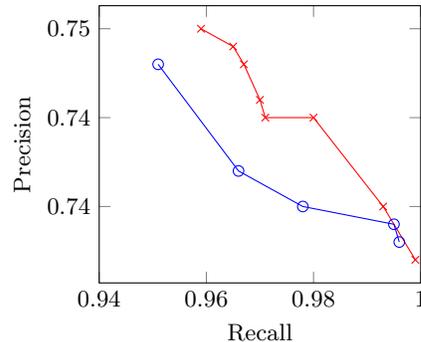
---

<sup>4</sup> We note that for this experiment we worked solely with nouns.

<sup>5</sup> Still, some of the values used produced equal results and thus are not visible in Figure 3.

| System | Nouns Nb. |          |
|--------|-----------|----------|
|        | FSc.      | Clusters |
| DEP    | 80.6      | 1.200    |
| LEX    | 79.5      | 1.257    |
| UOY    | 65.8      | 11.28    |

**Fig. 2.** Best F-scores obtained for both our methods on Task 02 of Semeval-2007, using lexical (LEX) and syntactic dependency (DEP) co-occurrences.



**Fig. 3.** Precision-Recall curve for LEX and DEP systems. To improve visibility, we focused the scale on the curves.

It must be noted that this particular Semeval task was dominated by the most frequent sense with an F-score of 80.7, assigning an average of one sense per target word. Our solutions assign an average of 1.257 and 1.200 for LEX and DEP respectively.

Based on our proof of concept experiment, we confirm that using syntactic dependencies in order to disambiguate word senses improves can improve the results when compared with regular lexical co-occurrence approaches.

## 6 Conclusion and future work

In this paper we analyzed the state of the art of linguistic network-based approaches to semantic similarity task from a graph-centric point of view. We reviewed the techniques in terms of its graph characteristics, from their structure to the algorithms employed. Among the literature covered, certain non-explored research paths were identified, namely the lack of syntactic data on the networks employed, and therefore, a homogeneous network nature that only allows for relations of a unique type.

We addressed with the proposition of a hypergraph linguistic model that is able to hold heterogeneous language information. We believe that this structure allows the integration multiple kinds of information and has vast potential in terms of which algorithms it can be used with. Our model was tested in a word sense induction proof of concept experiment and found interesting and encouraging results. Again, we note that the approach proposed to solve word sense disambiguation and induction is a proof of concept and as encouraging as the results are, we still need to improve the system in order to compete with the best solutions in the state of the art.

As future work, we are currently extending our algorithm to properly combine the different types of information within our model. We would like to test other

kind of graph inductions (instead of transforming the hypergraph into a bipartite graph), or even better, use the incidence matrix of the hypergraph to calculate custom similarity metrics. In this same context, we believe that a deep analysis on the semantic meaning of different types of similarities (and their magnitudes) between words is needed to better determine which metric to use in a specific context. Finally, we also plan to address other NLP domains with our hypergraph model, notably information extraction problems.

## References

1. Quillian, R.: Semantic memory. In: *Semantic Information Processing*. MIT Press (1968) 216–270
2. Choudhury, M., Mukherjee, A.: The structure and dynamics of linguistic networks. In Ganguly, N., Deutsch, A., Mukherjee, A., eds.: *Dynamics On and Of Complex Networks. Modeling and Simulation in Science, Engineering and Technology*. Birkhuser Boston (2009) 145–166
3. Miller, G.A.: Wordnet: A lexical database for english. *Commun. ACM* **38** (1995) 39–41
4. Navigli, R., Ponzetto, S.P.: Babelnet: Building a very large multilingual semantic network. In: *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*. (2010) 216–225
5. Navigli, R.: Word sense disambiguation: A survey. *ACM Comput. Surv.* **41** (2009) 10:1–10:69
6. Navigli, R.: A quick tour of word sense disambiguation, induction and related approaches. In Bielikov, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turn, G., eds.: *SOFSEM 2012: Theory and Practice of Computer Science*. Volume 7147 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2012) 115–129
7. Agirre, E., López de Lacalle, O., Soroa, A.: Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.* **40** (2014) 57–84
8. Mihalcea, R.F., Radev, D.R.: *Graph-based Natural Language Processing and Information Retrieval*. 1st edn. Cambridge University Press, New York, NY, USA (2011)
9. Mihalcea, R., Tarau, P., Figa, E.: Pagerank on semantic networks, with application to word sense disambiguation. In: *Proceedings of the 20th International Conference on Computational Linguistics. COLING '04, Stroudsburg, PA, USA, Association for Computational Linguistics* (2004)
10. Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: *Proceedings of the International Conference on Semantic Computing. ICSC '07, Washington, DC, USA, IEEE Computer Society* (2007) 363–369
11. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. (2007)
12. Navigli, R., Lapata, M.: Graph connectivity measures for unsupervised word sense disambiguation. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence. IJCAI'07, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.* (2007) 1683–1688
13. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. EACL '09, Stroudsburg, PA, USA, Association for Computational Linguistics* (2009) 33–41

14. Harris, Z.: Distributional structure. *Word* **10** (1954) 146–162
15. Firth, J.R.: A synopsis of linguistic theory 1930-55. **1952-59** (1957) 1–32
16. Vronis, J.: Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language* **18** (2004) 223 – 252
17. Klapaftis, I.P., Manandhar, S.: Uoy: A hypergraph model for word sense induction & disambiguation. In: Proceedings of the 4th International Workshop on Semantic Evaluations. SemEval '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 414–417
18. Navigli, R., Crisafulli, G.: Inducing word senses to improve web search result clustering. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. EMNLP '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 116–126
19. Qian, T., Ji, D., Zhang, M., Teng, C., Xia, C.: Word sense induction using lexical chain based hypergraph model. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, Dublin City University and Association for Computational Linguistics (2014) 1601–1611
20. Estrada, E., Rodriguez-Velazquez, J.A.: Complex networks as hypergraphs. arXiv preprint physics/0505137 (2005)
21. Hope, D., Keller, B.: 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), Atlanta, Georgia, USA, Association for Computational Linguistics (2013) 689–694
22. Liu, H., Le Pendu, P., Jin, R., Dou, D.: A hypergraph-based method for discovering semantically associated itemsets. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining. ICDM '11, Washington, DC, USA, IEEE Computer Society (2011) 398–406
23. Heintz, B., Chandra, A.: Beyond graphs: toward scalable hypergraph analysis systems. *ACM SIGMETRICS Performance Evaluation Review* **41** (2014) 94–97
24. Berge, C.: *Graphs and Hypergraphs*. Elsevier, Oxford, UK, UK (1985)
25. Ide, N., , Ide, N., Suderman, K.: The american national corpus first release. In: Proceedings of the Fourth Language Resources and Evaluation Conference (LREC. (2004) 1681–1684
26. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. (2014) 55–60
27. Agirre, E., Soroa, A.: Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In: Proceedings of the 4th International Workshop on Semantic Evaluations. SemEval '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 7–12