An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation

Roberto Navigli and Mirella Lapata

***Abstract*— Word sense disambiguation (WSD), the task of identifying the intended meanings (senses) of words in context, has been a long-standing research objective for natural language processing. In this paper we are concerned with graph-based algorithms for large-scale WSD. Under this framework, finding the right sense for a given word amounts to identifying the most “important” node among the set of graph nodes representing its senses. We introduce a graph-based WSD algorithm which has few parameters and does not require sense annotated data for training. Using this algorithm, we investigate several measures of graph connectivity with the aim of identifying those best suited for WSD. We also examine how the chosen lexicon and its connectivity influences WSD performance. We report results on standard data sets, and show that our graph-based approach performs comparably to the state of the art.**

***Index Terms*— word sense disambiguation, graph connectivity, semantic networks, social network analysis.**

I. Introduction

WORD sense disambiguation (WSD), the ability to identify the intended meanings of words (word senses) in context, is a central research topic in Natural Language Processing (NLP). Sense disambiguation is often characterized as an intermediate task, which is not an end in itself, but essential for many applications requiring broad-coverage language understanding. Examples include machine translation [1], information retrieval [2], question answering [3], and summarization [4].

Recent advances in WSD have benefited greatly from the availability of corpora annotated with word senses. Most accurate WSD systems to date exploit supervised methods which automatically learn cues useful for disambiguation from hand-labeled data. Although supervised approaches outperform their unsupervised alternatives (see [5], [6] for overviews), they often require large amounts of training data to yield reliable results [7], and their coverage is typically limited to the words for which sense labeled data exist. Unfortunately, creating sense tagged corpora manually is an expensive and labor-intensive endeavor [8] which must be repeated for new domains, languages, and sense inventories. Given the data requirements for supervised WSD and the current paucity of suitable data for many languages and text genres, unsupervised approaches would seem to offer near-term hope for large-scale sense disambiguation.

In the field of WSD, the term unsupervised is commonly used to describe approaches that perform sense disambiguation without resorting to labeled training data (see [9], [10]). Importantly, these approaches are not knowledge-free, since they are expected to disambiguate instances according to a pre-existing sense inventory and often exploit its structure and relations in order to perform the disambiguation task more accurately. A more restrictive view of “unsupervised” applies to methods for sense induction or discrimination, which attempt to automatically identify all possible senses of an ambiguous word without an inventory or labeled training data (see [11], [12]). The induced senses here have no external meaning, they only match natural divisions in the data with respect to the task.

Throughout this article we will use the term unsupervised to refer to knowledge-based WSD methods that employ an existing sense inventory but no labeled data (see also [13], [14] for a more detailed discussion of these issues). Most of these methods can be broadly divided in two categories, namely graph-based ones and similarity-based ones. Graph-based algorithms often consist of two stages [4], [9], [15]. First, a graph is built from a lexical knowledge base representing all possible interpretations of the word sequence being disambiguated. Graph nodes correspond to word senses, whereas edges represent dependencies between senses (e.g., synomymy, antonymy). Next, the graph structure is assessed to determine the importance of each node. Here, sense disambiguation amounts to finding the most “important” node for each word. Similarity-based algorithms assign a sense to an ambiguous word by comparing each of its senses with those of the words in the surrounding context [10], [16]. The sense whose definition has the highest similarity is assumed to be the correct one. The algorithms differ in the type of similarity measure they employ and the adopted definition of context which can vary from a few words to the entire corpus. In graph-based methods word senses are determined *collectively* by exploiting dependencies across senses, whereas in similarity-based approaches each sense is determined for each word *individually* without considering the senses assigned to neighboring words. Experimental comparisons between the two algorithm types (e.g., [9], [17]) indicate that graph-based algorithms outperform similarity-based ones, often by a significant margin.

In this paper we focus on graph-based methods and investigate in depth the role of graph structure in determining WSD performance. Specifically, we compare and contrast various measures of graph connectivity that assess the relative importance of a node within the graph. Graph theory is abundant with such measures and evaluations have been undertaken in the context of studying the structure of a hyperlinked environment [18] and within social network analysis [19]. Our experiments attempt to establish whether some of these measures are particularly appropriate for graph-based WSD. We also investigate the role of the chosen lexicon and its contribution to WSD. The inventory is of primary importance here as it determines the shape and structure of the smaller subgraphs upon which WSD takes place.

Manuscript received May 31, 2008. Revised Dec 3, 2008.

R. Navigli is with the Dipartimento di Informatica, University of Rome “La Sapienza”, 00198 Roma. Email: `navigli@di.uniroma1.it`. M. Lapata is with the School of Informatics, University of Edinburgh, EH8 9AB. Email: `mlap@inf.ed.ac.uk`.

Such a comparative study is novel; previous work restricts itself to a single lexicon and measure which is either devised specifically for WSD [4] or adopted from network analysis [9], [15]. Our contributions are three-fold: a general framework for graph-based WSD; an empirical comparison of a broad range of graph connectivity measures using standard evaluation data sets; and an investigation of the influence of the WordNet sense inventory and its graph structure on WSD.

II. Related Work

Measures of graph connectivity have been studied extensively in the social sciences, especially within the field of Social Network Analysis (SNA) [20]. A social network is a set of people or groups of people with some pattern of contacts or interactions between them. Examples include the patterns of friendship between individuals or the business relationships between companies. One of the fundamental problems in network analysis is to determine which individuals are most *central* or important in the network (by being most connected or having most influence) and how they are *connected* to one another. Quantifying centrality and connectivity allows us to characterize the structure and properties of large networks and to make predictions about their behavior (e.g., what happens if the network becomes more or less connected).

Recent years have witnessed great interest in network research, partly due to the expansion of the World Wide Web, and the development of link analysis algorithms for information retrieval. Among these, PageRank [21] and HITS [22] have been extremely influential. PageRank assigns a numerical weighting to each element of a hyperlinked set of documents, with the purpose of measuring its relative importance within the set, whereas HITS rates Web pages for their authority and hub values. Hubs and authorities exhibit a mutually reinforcing relationship: a good hub is a document that points to many others, and a good authority is a document that many documents point to (we discuss HITS and PageRank more formally in Section IV). Beyond information retrieval, link analysis algorithms have been applied in a variety of tasks. Examples include spam detection [23], topic-oriented crawling [24], keyword searching in relational databases [25] and measuring citation impact factor [26].

Graph-based approaches have also enjoyed growing popularity within NLP. This is because in many cases one is faced with the problem of selecting a single best candidate out of many interrelated objects. Word sense disambiguation is a case in point here. Assuming we have access to a dictionary which lists for each word its different senses, we can work out the multiple meanings of a word sequence (e.g., sentence, paragraph, document) by looking up the meaning of individual words in our dictionary. These different interpretations can be compactly represented as a graph where nodes correspond to senses and edges to sense relations (e.g., synonymy, hyponymy). Now, our task is to come up with a single sense for each ambiguous word in context. This can be done intuitively by selecting the sense with the most connections (i.e., incoming edges) in the graph [4], [27]. These connections can be weighted according to semantic type (e.g., synonymy relations are more important than hyponymy). In other work [15] senses are scored by taking edge paths into account. The PageRank algorithm has also been used to induce a ranking of the senses of an ambiguous word [9], [28]. Graph algorithms are appealing to WSD since they essentially work in an unsupervised setting without requiring data hand-labeled with correct word senses.

Graph algorithms have been also applied to word sense induction, the task of inferring automatically the senses of a given target word without recourse to a dictionary [29], [30]. For example, in the HyperLex algorithm [29] words are taken as the nodes of the graph and word co-occurrence represents an edge between two nodes. Detecting the different senses of a word thus amounts to isolating the high-density components in this co-occurrence graph. Although we focus here primarily on unsupervised methods, it is worth pointing out that graph algorithms such as Label Propagation [31] have been successfully employed in supervised WSD [32]. Beyond WSD, graph-based methods have been adopted in many NLP tasks such as summarization [33]–[35], keyword extraction [34], sentiment analysis [36], sentence retrieval for question answering [37], ontology learning [38], human conversation analysis [39], and for estimating word dependency distributions [40].

Despite the popularity of graph-based methods in NLP, there have been virtually no studies assessing how graph connectivity and the different ways of measuring it affect different tasks. A large number of graph connectivity metrics have been proposed within social network analysis and applied to different networks. Recent research shows that there is no single universally appropriate metric [41]. Previous work has used almost exclusively two metrics, either variants of degree centrality [4], [27] or PageRank [9], [28]. This is in marked contrast with similarity-based approaches, where several studies have evaluated the effect of similarity measures on WSD performance [42], [43]. Another related issue concerns the dictionary employed for constructing the sense graph. The latter shapes the topology of the graph and determines its connectivity patterns. For instance, a densely connected graph will be created from a dictionary that lists many sense relations. Our own work [44] has explored some of these issues, in a rather restricted setting. Specifically, we used the graph algorithm presented in [15] to build the sentence representations used to assess the performance of graph connectivity measures. The algorithm builds the sense graph by consulting a hand constructed grammar that determines which graph edge sequences are valid. Unfortunately, this casts doubt on the generalizability of our results since the connectivity measures are applied to an idealized graph with only meaningful edges. So, it is not clear whether differences in WSD performance are due to a specific connectivity measure, to the graph construction process, or to their interaction.

In this paper we analyze the impact of connectivity metrics for unsupervised WSD on a sounder empirical and theoretical footing. We devise a general-purpose graph-based algorithm that does not rely on a hand constructed grammar. We analyze its computational complexity depending on the connectivity measure of choice and provide a detailed study on how WordNet and its relations (number and type) affect WSD performance. To preview our results, we find that Degree, a relatively simple measure, outperforms more sophisticated alternatives and delivers state-of-the-art performance. We also find that lexicons with many connections between senses are beneficial for graph-based WSD. In all cases, we present results on benchmark data sets in the context of a competitive baseline algorithm [45] (contrary to [44], where we compare against a naive baseline that selects a sense for each word at random).

III. Graph-based WSD

In order to isolate the impact of graph connectivity measures on WSD, we devised a fairly general disambiguation algorithm that has few parameters and relies almost exclusively on graph structure for inferring word senses. In common with much current work in WSD, we are assuming that meaning distinctions are provided by a reference lexicon, which encodes for each word a discrete set of senses. Although our experiments will use the WordNet sense inventory [46], neither our graph-based algorithm nor the proposed connectivity measures are limited to this particular lexicon. Resources with alternative sense distinctions and structure could also serve as input to our method. In the following we first provide a brief introduction to WordNet. Next, we describe our WSD algorithm and show a working example.

A. The Lexicon

WordNet is an online lexical reference system[1] whose design is inspired by psycholinguistic theories of human lexical memory [46]. The WordNet lexicon contains nouns, verbs, adjectives, and adverbs. Lexical information is organized in terms of word meanings, rather than word forms. Senses in the WordNet database are represented relationally by synonym sets (henceforth *synsets*) — the sets of all words sharing a common sense. As an example consider three senses of the verb *drink*, "consume liquids", "consume alcohol" and "toast". These are respectively represented as:

(1) a. $\{\textit{drink}_v^1,\ \textit{imbibe}_v^3\}$
b. $\{\textit{drink}_v^2,\ \textit{booze}_v^1,\ \textit{fuddle}_v^2\}$
c. $\{\textit{toast}_v^2,\ \textit{drink}_v^3,\ \textit{pledge}_v^2,\ \textit{salute}_v^1,\ \textit{wassail}_v^2\}$

Each word in a synset is associated with a part of speech which we denote with a subscript: n stands for noun, v for verb, a for adjective, and r for adverb. The superscript denotes the sense number associated with each word (e.g., $\textit{drink}_v^2$ corresponds to the second sense of the verb *drink*). Each synset is associated with a gloss, i.e., a textual definition which explains its meaning. For example, the synset in (1-b) is defined as "consume alcohol". Moreover, the synsets for each word are ranked according to their frequency of occurrence in the SemCor corpus [47] which is a subset of the Brown corpus annotated with word senses (see Section VI-A for details). Thus, the first sense given for a word in WordNet is attested more times in SemCor than the second one, which in turn is more frequent than the third one, etc. The latest WordNet version (3.0) contains approximately 155,000 words organized in over 117,000 synsets.

WordNet also encodes lexical and semantic relations. The former connect pairs of word senses, whereas the latter relate synsets. Lexical relations in WordNet are nominalization (e.g., the noun $\textit{drinking}_n^1$ is a nominalization of the verb $\textit{drink}_v^1$), antonymy (e.g., $\textit{cold}_a^1$ is an antonym of $\textit{hot}_a^1$), pertainymy (e.g., $\textit{dental}_a^1$ pertains to $\textit{tooth}_n^1$), and so on. Examples of semantic relations are hypernymy (e.g., $\{\textit{milk}_n^1\}$ is a kind of $\{\textit{beverage}_n^1,\ \textit{drink}_n^3,\ \textit{drinkable}_n^1,\ \textit{potable}_n^1\}$) and meronymy (e.g., $\{\textit{milk}_n^1\}$ has-part $\{\textit{protein}_n^1\}$). Notice that we can transform a lexical relation into a semantic relation by extending the relation existing between a pair of word senses to the synsets which contain them. For example, we can extend the nominalization relation in (2-a) to the two synsets containing the senses $\textit{drinking}_n^1$ and $\textit{drink}_v^1$ (see (2-b)):

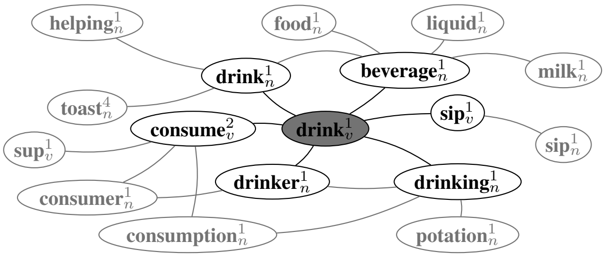


Fig. 1. An excerpt of the WordNet graph centered around $\textit{drink}_v^1$.

(2) a. $\textit{drinking}_n^1 \xrightarrow{\text{NOM}} \textit{drink}_v^1$
b. $\{\textit{drinking}_n^1,\ \textit{imbibing}_n^1,\ \textit{imbibition}_n^2\} \xrightarrow{\text{NOM}} \{\textit{drink}_v^1,\ \textit{imbibe}_v^3\}$

where NOM denotes the nominalization relation.

We can view WordNet as a graph whose nodes are synsets and edges lexical and semantic relations between synsets. Even though WordNet does not include a gloss relation, we can induce it heuristically [48] for a pair of synsets S and S' connected via a gloss relation if an unambiguous word $w \in S'$ occurs in the gloss of S.[2] Note that w must be unambiguous, otherwise S should have been connected with the appropriate sense of w. For example, the first sense of $\textit{milk}_n$ is defined as "a white nutritious liquid secreted by mammals and used as food by human beings". Here, the words $\textit{nutritious}_a$, $\textit{mammal}_n$ and $\textit{human being}_n$ have only one sense in WordNet, so we can infer the following gloss relations:

(3) a. $\textit{milk}_n^1 \xrightarrow{\text{GLOSS}} \textit{nutritious}_a^1$
b. $\textit{milk}_n^1 \xrightarrow{\text{GLOSS}} \textit{mammal}_n^1$
c. $\textit{milk}_n^1 \xrightarrow{\text{GLOSS}} \textit{human being}_n^1$

In Figure 1 we show an excerpt of the WordNet graph centered around the synset $\{\textit{drink}_v^1,\ \textit{imbibe}_v^3\}$. In this graph, nodes correspond to synsets which we abbreviate to a single word in the synset (e.g., $\textit{drink}_v^1$ corresponds to $\{\textit{drink}_v^1,\textit{imbibe}_v^3\}$). The node for $\textit{drink}_v^1$ is drawn as a dark gray ellipse, whereas adjacent vertices (senses) are shown as white ellipses (e.g., $\textit{consume}_v^2$, $\textit{beverage}_n^1$). In graph theory, an adjacent vertex[3] of a graph is a vertex that is connected to another vertex with an edge within the graph. In our case, adjacent vertices represent relations in WordNet. Senses which are not directly adjacent to $\textit{drink}_v^1$, but reachable through a sequence of edges are shown in light gray (e.g., $\textit{toast}_n^4$, $\textit{milk}_n^1$). Notice that the graph is undirected and does not explicitly encode the different kinds of relations.

B. The WSD Algorithm

Our disambiguation algorithm proceeds incrementally on a sentence-by-sentence basis. Initially, we build a graph $G = (V, E)$ for each target sentence σ which we induce from the graph of the reference lexicon. Sentences are part-of-speech tagged, our algorithm considers content words only (i.e., nouns, verbs, adjectives, and adverbs). As explained in Section II the nodes in the graph are word senses and the edges semantic relations.

[1] WordNet is available from `http://wordnet.princeton.edu`

[2] WordNet has recently released sense disambiguated glosses (`http://wordnet.princeton.edu/glosstag`) which we could use instead of the heuristic approach described above. However, these are specific to English, and our aim is to be as language neutral as possible.

[3] Throughout this article we use the terms "vertex" and "node" interchangeably.

Given this graph G, we select for each content word $w_i \in \sigma$ the most appropriate sense $S_{w_i} \in Senses(w_i)$, where $Senses(w_i)$ is the set of senses of w_i listed in WordNet. We accomplish this by ranking each vertex in the graph G according to its importance which we operationalize in terms of graph connectivity.

More formally, given a word sequence $\sigma = (w_1, w_2, \ldots, w_n)$, we perform the following steps to construct G:

1) Let $V_\sigma := \bigcup_{i=1}^{n} Senses(w_i)$ denote all possible word senses in σ. We set $V := V_\sigma$ and $E := \emptyset$.
2) For each node $v \in V_\sigma$, we perform a depth-first search (DFS) of the WordNet graph: every time we encounter a node $v' \in V_\sigma$ ($v' \neq v$) along a path $v, v_1, \ldots, v_k, v'$ of length $\leq L$, we add all intermediate nodes and edges on the path from v to v': $V := V \cup \{v_1, \ldots, v_k\}$ and $E := E \cup \{\{v, v_1\}, \ldots, \{v_k, v'\}\}$.

In DFS, edges are explored out of the most recently discovered vertex v that still has unexplored edges leaving it. When all of v's edges have been explored, the search "backtracks" to explore edges leaving the vertex from which v was discovered. This process continues until we have discovered all the vertices that are reachable from the original source vertex. Our use of depth-first search is motivated by computational efficiency. However, there is nothing inherent in our formulation that restricts us to this graph traversal algorithm. For instance, we could have adopted breadth-first search (BFS) which has been previously employed in graph-based WSD [49].

Let us illustrate our graph construction process with a simple example. Consider the sentence *She drank some milk*. Here, the content words are *drink* and *milk* (i.e., $\sigma = (drink_v,\ milk_n)$) for which WordNet lists 5 and 4 senses, respectively (we omit $some_a$ from the example for the sake of brevity). Initially we set $V_\sigma := \{drink_v^1, \ldots, drink_v^5,\ milk_n^1, \ldots, milk_n^4\}$, $V := V_\sigma$, and $E := \emptyset$. Next, we perform a DFS from the vertex $drink_v^1$. In WordNet, this vertex is adjacent to $drink_n^1$ and $drinker_n^2$ (via a nominalization relation), and to $beverage_n^1$ (via a gloss relation). We first follow $drink_n^1$, which is in turn connected to $beverage_n^1$. The latter is adjacent to $milk_n^1$, which is a sense in V_σ. Consequently, we add to the graph all edges and vertices in the path between $drink_v^1$ and $milk_n^1$: $V := V \cup \{drink_n^1, beverage_n^1\}$, and $E := E \cup \{\{drink_v^1, drink_n^1\}, \{drink_n^1, beverage_n^1\}, \{beverage_n^1, milk_n^1\}\}$ (see Figure 2a, new vertices and edges are highlighted in bold).

When the search backtracks to $beverage_n^1$, another path can be followed leading to $milk_n^2$. We therefore set $V := V \cup \{food_n^1, nutriment_n^1\}$, and $E := E \cup \{\{beverage_n^1, food_n^1\}, \{food_n^1,\ nutriment_n^1\}, \{nutriment_n^1, milk_n^2\}\}$ (see Figure 2b). The search next backtracks to $drink_v^1$, so a new adjacent vertex can be followed. As mentioned earlier, $drink_v^1$ is also connected to $beverage_n^1$ (besides $drink_n^1$), so we add the edge $\{drink_v^1, beverage_n^1\}$ to E (Figure 2c). Analogously, a new path from $drink_v^1$ passes through $drinker_n^2$ and $beverage_n^1$, leading to the following update: $V := V \cup \{drinker_n^2\}$, and $E := E \cup \{\{drink_v^1, drinker_n^2\}, \{drinker_n^2,\ beverage_n^1\}\}$ (Figure 2d).

At this point the DFS from $drink_v^1$ is completed as we cannot find any more edge paths connecting it to other senses in V_σ. We next perform a new DFS from $drink_v^2$. The latter is adjacent in WordNet to $drinker_n^2$ and $boozing_n^1$. Since we have already created a vertex for $drinker_n^2$ (Figure 2d), we simply add the edge $\{drink_v^2, drinker_n^2\}$ to E (Figure 2e) and create a new vertex for $boozing_n^1$ (Figure 2f) which in turn is connected to $beverage_n^1$ (through the gloss relation). The search now stops, as $beverage_n^1$ has been visited before, and the graph is updated: $V := V \cup \{boozing_n^1\}$, and $E := E \cup \{\{drink_v^2, boozing_n^1\}, \{boozing_n^1,\ beverage_n^1\}\}$.

The DFS does not find any related senses for $drink_v^3$ and $drink_v^4$, so we move on to $drink_v^5$. This sense is adjacent to $drinker_n^2$ and $boozing_n^1$ in WordNet. As both vertices have been visited before, we add the edges $\{drink_v^5, drinker_n^2\}$, and $\{drink_v^5, boozing_n^1\}$ to E (Figures 2g and 2h). The graph construction procedure now terminates, as there are no more paths connecting senses in V_σ. The final graph represents the different meanings of the sentence *She drank some milk* ($3 \cdot 2 = 6$ interpretations in total). Merely by inspection we can see that the graph is relatively dense. Each vertex is ≤ 3 edges apart from some vertex in the original set V_σ of word senses (where 3 is half the maximum length of any path). For instance, $beverage_n^1$ is at distance one from $milk_n^1$, whereas $food_n^1$ is at distance two from $milk_n^1$ and $milk_n^2$.

Given this graph, we must next evaluate which one of the meanings of $drink_v$ and $milk_n$ (see the dark gray ellipses in Figure 2) is most important. Many measures of graph connectivity can be used to accomplish this, we provide a detailed discussion in the following section. Here, we briefly note that these measures can be either *local* or *global*. Local measures capture the degree of connectivity conveyed by a single vertex in the graph towards all other vertices, whereas global measures estimate the overall degree of connectivity of the entire graph.

Arguably, the choice of connectivity measure influences the selection process for the highest-ranked sense. Given a *local measure* l, and the set of vertices V_σ, we induce a ranking of the vertices $rank_l$ such that $rank_l(v) \leq rank_l(v')$ iff $l(v) \geq l(v')$. Then, for each word $w_i \in \sigma$, we select the best-ranking sense in $Senses(w_i)$ according to $rank_l$. The ranking function here operates on the entire set of vertices V_σ and is thus realized for all the senses of all words at once. Alternatively, we could define the ranking function on the senses for each word. The former formulation represents the gist of the sentence rather than the meaning of isolated words, thus highlighting those concepts that are more likely to convey core meaning. A *global measure* g characterizes the overall graph structure G with a single number and is thus not particularly helpful in selecting a unique sense for ambiguous words – G collectively represents all interpretations of σ. We get around this problem, by applying g to each interpretation of σ and selecting the highest scoring one. An interpretation is a subgraph $G' \subseteq G$ such that G' includes one and only one sense of each word in sentence σ and all their corresponding intermediate nodes. So, if our sentence has twenty interpretations, we will measure the connectivity of the twenty resulting subgraphs and choose the best-ranking one.

The disambiguation algorithm presented above has a limited notion of context, only neighboring words within the same sentence contribute to the meaning of an ambiguous word. It thus differs from [9], [27] who build a disambiguation graph for an entire document and assign the same sense to all occurrences of a word. In our case, the senses of a word can vary across sentences and documents. There is nothing inherent in our algorithm that restricts us to sentences. We could just as well build and disambiguate a graph for a document. We sacrificed a small amount of accuracy — previous work [50] shows that

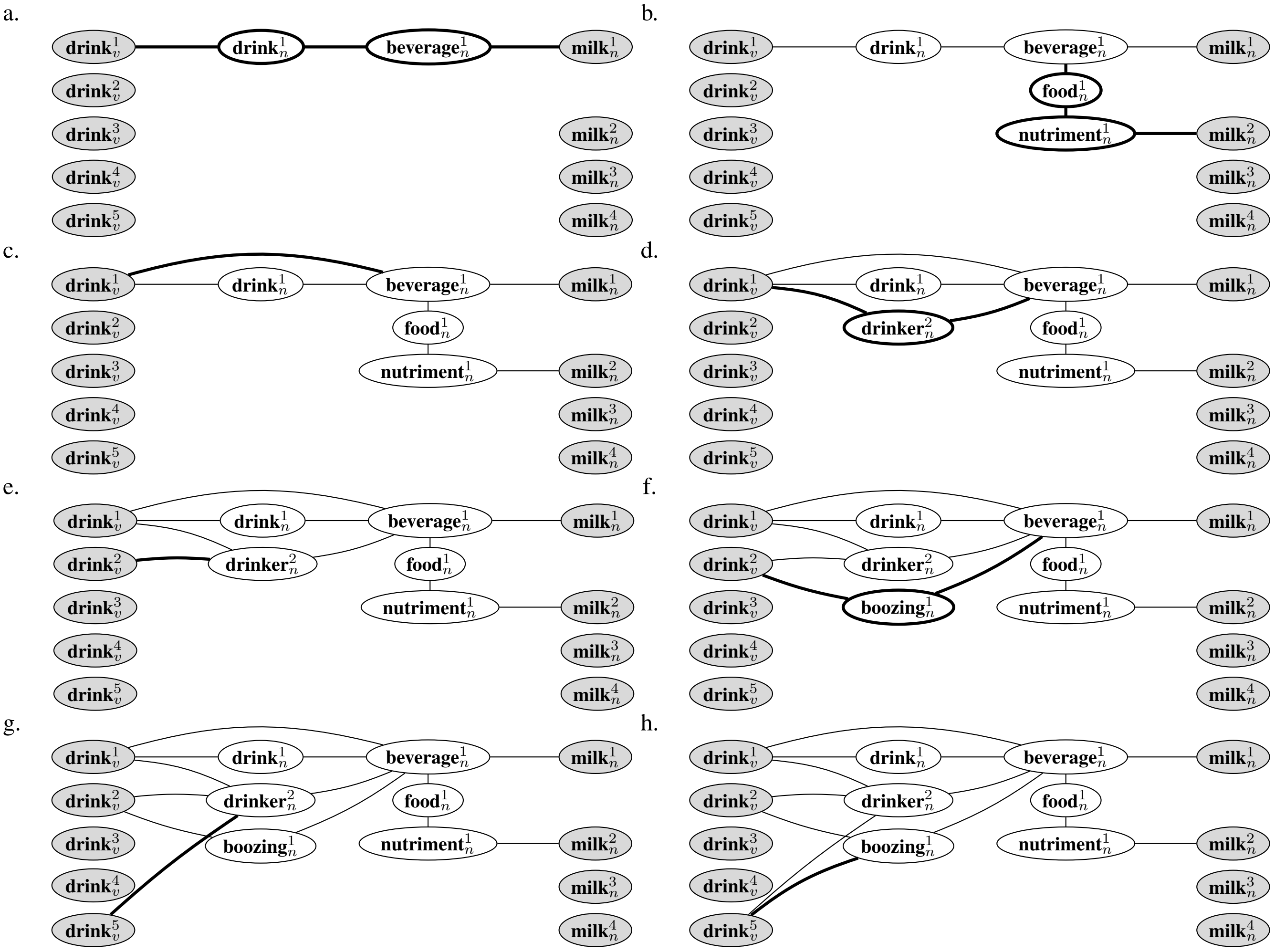


Fig. 2. Graph construction process for the sentence *She drank some milk* (σ = (drink$_v$, milk$_n$)).

polysemous words appearing two or three times in the same discourse tend to share the same sense — to gain efficiency (as we shall see below disambiguating with global connectivity measures is computationally demanding). We also depart from [9], [27] in our use of unweighted unlabeled edges. The reasons for this are twofold. First, there is no agreed upon method for inferring edge weights (these are set by hand in [4], [27] and determined by gloss overlap in [9]). Secondly, aside from the problem of computing the weight of an edge, which warrants a separate study on its own, we wanted to isolate the influence of graph connectivity metrics on WSD without any potentially confounding factors. Finally, we should point out that our algorithm can be used without making use of the first sense information available in WordNet. The latter is often used (e.g., [27]) to break ties, however we default to random choice unless otherwise stated. The sense frequencies in WordNet are derived from the hand-labeled SemCor corpus (see Section III-A). Albeit valuable, these are not typically available in other languages or domains.

IV. Connectivity Measures

In this section we describe the measures of graph connectivity we consider for unsupervised WSD. Although our measures can be applied to both directed and undirected graphs, in the context of WSD we are assuming that we are dealing with undirected graphs. This is motivated by the fact that semantic relations often have an inverse counterpart (e.g., hypernymy is the inverse relation of hyponymy). Moreover, the assumption enables the use of knowledge resources which do not explicitly specify relation directions or provide some connections without an inverse counterpart.

We first introduce the distance function $d(u,v)$, which is used by some of the measures discussed below:

$$d(u,v) = \begin{cases} \text{length of shortest path} & \text{if } u \rightsquigarrow v \\ K & \text{otherwise} \end{cases} \qquad (1)$$

where $u \rightsquigarrow v$ indicates the existence of a path from u to v, and K is a conversion constant [18], which replaces the ∞ distance with an integer when v is not reachable from u (we choose $K = |V|$, as the length of any shortest path is $< |V|$). The length of a path is calculated as the number of edges in the path. For example, in Figure 2h, $d(drink_v^1, milk_n^2) = 4$ and $d(drink_v^1, milk_n^1) = 2$.

A. Local Measures

Local measures of graph connectivity determine the degree of relevance of a single vertex v in a graph G. They can thus be viewed as measures of the influence of a node over the network. Formally, we define a local measure l as:

$$l : V \rightarrow [0,1] \qquad (2)$$

A value close to one indicates that a vertex is important, whereas a value close to zero indicates that the vertex is peripheral.

Several local measures of graph connectivity have been proposed in the literature (see [20] for a comprehensive overview). Many of these rely on the notion of *centrality*: a node is central if it is maximally connected to all other nodes. In the following, we consider three best-known measures of centrality, namely degree, closeness, and betweenness [51], and variants thereof.

1) Degree Centrality: The simplest way to measure vertex importance is by its degree, i.e., the number of edges terminating in a given vertex:

$$deg(v) = |\{\{u, v\} \in E : u \in V\}| \tag{3}$$

A vertex is central, if it has a high degree. Conversely, a disconnected vertex has degree zero. Degree centrality is the degree of a vertex normalized by the maximum degree:

$$C_D(v) = \frac{deg(v)}{|V| - 1} \tag{4}$$

According to the graph in Figure 2h, $C_D(drink_v^1) = \frac{3}{14}$, $C_D(drink_v^2) = C_D(drink_v^5) = \frac{2}{14}$, and $C_D(milk_n^3) = C_D(milk_n^4) = 0$. So, the sense for *drink*$_v$, with the highest degree score, is *drink*$_v^1$, whereas there is a tie for *milk*$_n$. As discussed earlier (see Section III-B), in this case we choose randomly. The degree centrality scores for all graph nodes are given in Table I. We show the best scores for the senses of *drink*$_v$ and *milk*$_n$ in bold face. The score for disconnected nodes is zero (e.g., *drink*$_v^3$).

2) Eigenvector Centrality: A more sophisticated version of degree centrality is eigenvector centrality. Whereas the former gives a simple count of the number of connections a vertex has, the latter acknowledges that not all connections are equal. It assigns relative scores to all nodes in the graph based on the recursive principle that connections to nodes having a high score contribute more to the score of the node in question [52]. The term eigenvector centrality stems from the fact that it calculates the dominant eigenvector of a matrix associated with (and possibly equal to) the adjacency matrix of the target graph. PageRank [21] and HITS [22] are variants of the eigenvector centrality measure and have been a popular choice in graph-based WSD (see Section II).

PageRank determines the relevance of a node v recursively based on a Markov chain model. All nodes that link to v contribute towards determining its relevance. Each contribution is given by the page rank value of the respective node ($PR(u)$) divided by the number of its neighbors:

$$PR(v) = \frac{(1-\alpha)}{|V|} + \alpha \sum_{\{u,v\} \in E} \frac{PR(u)}{outdegree(u)} \tag{5}$$

The overall contribution is weighted with a damping factor α, which implements the so-called random surfer model: with probability $1 - \alpha$, the random surfer is expected to discontinue the chain and select a random node, with relevance $\frac{1}{|V|}$. Thus, if a node is disconnected, its PageRank value is given by the first term in equation (5) (this value is small – close to zero – when V is large).

In contrast, HITS (Hypertext Induced Topic Selection) determines two values for each node v, the authority ($a(v)$) and the hub value ($h(v)$). In a directed graph these are defined in terms of one another in a mutual recursion:

$$h(v) = \sum_{u:(u,v) \in E} a(u) \quad ; \quad a(v) = \sum_{u:(v,u) \in E} h(u) \tag{6}$$

Intuitively, a good hub is a node that points to many good authorities, whereas a good authority is a node that is pointed to by many good hubs. The hub and authority values of a disconnected node are both zero.

A major difference between HITS and PageRank is that the former is computed dynamically on a subgraph of relevant pages, whereas the latter takes the entire graph structure into account.

Both algorithms iteratively calculate their ranking scores through increasingly precise approximations. In Figure 2h, the authority values for the nodes representing the senses of *drink*$_v$ and *milk*$_n$ are $a(drink_v^1) = 0.39$, $a(drink_v^2) = a(drink_v^5) = 0.23$, $a(milk_n^1) = 0.17$, and $a(milk_n^2) = 0.02$. The PageRank values for these nodes are $PR(drink_v^1) = 0.08$, $PR(drink_v^2) = PR(drink_v^5) = 0.05$, and $PR(milk_n^1) = PR(milk_n^2) = 0.03$. HITS unequivocally selects *drink*$_v^1$ and *milk*$_n^1$ as the best senses for our sentence. PageRank agrees with HITS on the sense assignment for *drink*$_v$, but yields a tie for *milk*$_n$ (again to be resolved by random selection). Generally, HITS tends to deliver finer grained sense rankings than PageRank. In Figure 2h, ranking all nodes with HITS delivers ten distinct non-zero scores, whereas PageRank only six (see Table I). Note that, since our graphs are undirected, the authority and hub values coincide. In fact, the edges (u, v) and (v, u) collapse to a single undirected edge $\{u, v\}$.

3) Key Player Problem (KPP): With KPP, a vertex is considered important if it is relatively close to all other vertices [53]:

$$KPP(v) = \frac{\displaystyle\sum_{u \in V : u \neq v} \frac{1}{d(u,v)}}{|V| - 1} \tag{7}$$

where the numerator is the sum of the inverse shortest distances between v and all other nodes and the denominator is the number of nodes in the graph (excluding v). The KPP of a disconnected node is a small constant, given by $\frac{1}{K} = \frac{1}{|V|}$.

For example, in Figure 2h $KPP(drink_v^1) = \frac{3+5\cdot\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+4\cdot\frac{1}{14}}{14} = 0.45$. KPP selects *drink*$_v^1$ and *milk*$_n^1$ as the best scoring senses for our sentence (see Table I).

KPP is similar to the better known closeness centrality measure [54] which is defined as the reciprocal of the total shortest distance from a given node to all other nodes. Here, we consider only KPP since it outperformed closeness centrality in our experiments.

4) Betweenness Centrality: The betweenness of vertex v is calculated as the fraction of shortest paths between node pairs that pass through v [51]. Formally, betweenness is defined as:

$$betweenness(v) = \sum_{s,t \in V : s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{8}$$

where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ the number of shortest paths from s to t that pass through vertex v. We normalize by dividing $betweenness(v)$ by the maximum number of node pairs excluding v:

$$C_B(v) = \frac{betweenness(v)}{(|V|-1)(|V|-2)} \tag{9}$$

The intuition behind betweenness is that a node is important if it is involved in a large number of paths compared to the total set of paths. The betweeness of a disconnected node is zero as no path can pass through it.

In Figure 2h, the vertices $\{drink_v^2, beverage_n^1\}$ are connected by two shortest paths, $drink_v^2, drinker_n^2, beverage_n^1$ and

Node	Degree	KPP	HITS	PR	Betw.
drink$_v^1$	**0.21**	**0.45**	**0.39**	**0.08**	**0.013**
drink$_v^2$	0.14	0.37	0.23	0.05	0.005
drink$_v^3$	0.00	0.07	0.00	0.00	0.000
drink$_v^4$	0.00	0.07	0.00	0.00	0.000
drink$_v^5$	0.14	0.37	0.23	0.05	0.005
drink$_n^1$	0.14	0.39	0.29	0.05	0.000
drinker$_n^2$	0.29	0.49	0.43	0.10	0.067
boozing$_n^1$	0.21	0.45	0.31	0.08	0.054
beverage$_n^1$	0.43	0.58	0.55	0.16	0.157
food$_n^1$	0.14	0.42	0.19	0.06	0.057
nutriment$_n^1$	0.14	0.35	0.07	0.06	0.033
milk$_n^1$	**0.07**	**0.36**	**0.17**	**0.03**	**0.000**
milk$_n^2$	**0.07**	0.27	0.02	**0.03**	**0.000**
milk$_n^3$	0.00	0.07	0.00	0.00	**0.000**
milk$_n^4$	0.00	0.07	0.00	0.00	**0.000**

TABLE I

VALUES OF LOCAL CONNECTIVITY MEASURES FOR THE NODES IN FIGURE 2(H). THE BEST SCORES FOR THE SENSES OF *drink*$_v$ AND *milk*$_n$ FOR EACH CONNECTIVITY MEASURE ARE SHOWN IN BOLD FACE.

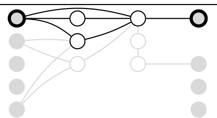
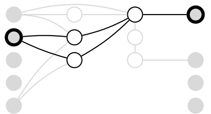
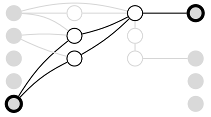
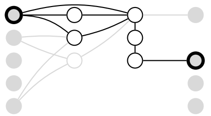
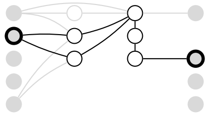
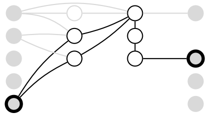
Interpretation	Compactness	Entropy	Edge Density
1. 	**0.90**	0.94	**0.60**
2. 	0.85	0.97	0.50
3. 	0.85	0.97	0.50
4. 	0.82	0.96	0.38
5. 	0.79	**0.98**	0.33
6. 	0.79	**0.98**	0.33

TABLE II

GLOBAL MEASURES APPLIED TO 6 GRAPHS REPRESENTING THE MEANINGS OF *She drank some milk* ACCORDING TO WORDNET.

$drink_v^2, boozing_n^1, beverage_n^1$. Thus, $\sigma_{drink_v^2, beverage_n^1} = 2$ and $\sigma_{drink_v^2, beverage_n^1}(drinker_n^2) = 1$. Once we obtain all σ values for *drinker*$_n^2$, we can calculate $betweenness(drinker_n^2) = 12.19$ and $C_B(drinker_n^2) = \frac{12.19}{14 \cdot 13} = 0.067$. As shown in Table I, *drink*$_v^1$ has the highest betweenness score for *drink*$_v$. The senses of *milk*$_n$ score 0, since they are not an intermediate node in any path between the senses of *drink*$_v$ and *milk*$_n$.

B. Global Measures

Global connectivity measures are concerned with the structure of the graph as a whole rather than with individual nodes. Here, we discuss three well-known measures, namely compactness, graph entropy, and edge density.

1) Compactness: This measure represents the extent of cross referencing in a graph [18]: when compactness is high, each vertex can be easily reached from other vertices. The measure is defined as:

$$CO(G) = \frac{Max - \sum_{u \in V} \sum_{v \in V} d(u, v)}{Max - Min} \tag{10}$$

where $Max = K \cdot |V|(|V| - 1)$ is the maximum value the distance sum can assume (for a completely disconnected graph) and $Min = |V|(|V| - 1)$ the minimum value (for a fully connected graph). $CO(G)$ is zero when G is completely disconnected and one when G is a complete graph.

Table II illustrates the compactness scores for the different interpretations (i.e. sense assignments) of our example sentence. For instance, the compactness of graph (1) is $CO(G) = \frac{(5 \cdot 5 \cdot 4) - 28}{(5 \cdot 5 \cdot 4) - (5 \cdot 4)} = \frac{72}{80} = 0.90$ (where $K = |V| = 5$). As we can see, this graph obtains the highest score among all possible sense interpretations (here, *drink*$_v$ disambiguates to *drink*$_v^1$ and *milk*$_n$ to *milk*$_n^1$).

2) Graph Entropy: Entropy measures the amount of information (or, alternatively, uncertainty) in a random variable. In graph-theoretic terms, high entropy indicates that many vertices are equally important, whereas low entropy indicates that only a few vertices are relevant. We define a simple measure of graph entropy as:

$$H(G) = -\sum_{v \in V} p(v) log(p(v)) \tag{11}$$

where the vertex probability $p(v)$ is determined by the degree distribution $\left\{ \frac{deg(v)}{2|E|} \right\}_{v \in V}$. To obtain a measure with a $[0, 1]$ range (0 for a totally disconnected graph, 1 for a complete graph), we divide $H(G)$ by the maximum entropy given by $\log |V|$. For example, the distribution associated with graph (1) in Table II is $(\frac{3}{12}, \frac{2}{12}, \frac{4}{12}, \frac{2}{12}, \frac{1}{12})$ (with $|E| = 6$) yielding an overall graph entropy $H(G) = \frac{2.18}{\log 5} = 0.94$. Graphs (5) and (6) have the highest entropy in Table II.

3) Edge Density: Finally, we propose the use of edge density as a simple global connectivity measure. Edge density is calculated as the ratio of edges in a graph to the number of edges of a complete graph with $|V|$ vertices (given by $\binom{|V|}{2}$). Formally:

$$ED(G) = \frac{|E(G)|}{\binom{|V|}{2}} \tag{12}$$

$ED(G)$ has a $[0, 1]$ range, with 0 corresponding to a totally disconnected graph and 1 to a complete graph. For example, graph (1) in Table II has edge density $ED(G) = \frac{6}{\binom{5}{2}} = \frac{6}{10} = 0.60$, and graph (2) $ED(G) = \frac{5}{\binom{5}{2}} = \frac{5}{10} = 0.50$.

The use of global connectivity measures makes our WSD algorithm susceptible to combinatorial explosion, since all possible interpretations of a given sentence must be ranked (see the complexity analysis in Section V). There are several heuristic search methods to solve combinatorial optimization problems [55]. Examples include local search, simulated annealing and genetic algorithms. For our WSD problem, we would like to adopt a search strategy that obtains a reasonable interpretation, even though it is not always the best possible one. Although a detailed study of heuristic search algorithms falls outside the scope of this article, we nevertheless compared two widely used methods, namely simulated annealing (SA) and genetic algorithms (GAs). SA has been previously applied to WSD [56], while the use of GAs is novel to our knowledge. In the following we briefly sketch how we adapt these algorithms to our task. We report on their performance in Section VI. In our setting, the search space consists of interpretations $\mathcal{I} = (S_1, \ldots, S_n)$ for a sentence $\sigma = (w_1, \ldots, w_n)$ such that $S_i \in Senses(w_i)$ for all $i = 1, \ldots, n$.

a) Simulated Annealing: Initially, we randomly select an interpretation $\mathcal{I}$ for sentence σ. At each step, we (randomly) select a word from σ and assign it a new sense also chosen at random. As a result, a new interpretation $\mathcal{I}'$ is produced. Next, we apply our global measure to the graph induced by $\mathcal{I}'$ and calculate the difference (ΔE) between its value and that of the graph obtained from the old interpretation $\mathcal{I}$. If the new interpretation has a higher score, we adopt it (i.e., we set $\mathcal{I} := \mathcal{I}'$). Otherwise, we either retain the old interpretation with probability $1 - e^{\frac{\Delta E}{T}}$ or nonetheless switch to the new interpretation with probability $e^{\frac{\Delta E}{T}}$, where T is a constant. The procedure is repeated u times. The algorithm terminates when we observe no changes in $\mathcal{I}$ after u steps. Otherwise, the entire procedure is repeated starting from the most recent interpretation.

b) Genetic Algorithms: Our genetic algorithm [57] starts from an initial random population P of p individuals (i.e., sentence interpretations) and iteratively creates a new population P' from a previous generation P by performing the following three steps:

i) probabilistically select $(1 - r) \cdot p$ elements of P to be kept in P' ($r \in [0, 1]$);
ii) probabilistically select $\frac{r \cdot p}{2}$ pairs from P, apply a crossover operator to each pair, and add the resulting pair of individuals to P'. We adopted single-point crossover which, given two individuals $(S_1, \ldots, S_n)$ and $(S'_1, \ldots, S'_n)$ establishes a crossover point $i \in \{1, \ldots, n-1\}$ and generates two offsprings $(S_1, \ldots, S_i, S'_{i+1}, \ldots, S'_n)$ and $(S'_1, \ldots, S'_i, S_{i+1}, \ldots, S_n)$;
iii) mutate $m \cdot p$ individuals in P' ($m \in [0, 1]$). Mutation consists of the random change of one sense in the chosen individuals.

Probabilistic selection is performed by evaluating the fitness of each individual in the population. Our fitness function is the graph connectivity measure (e.g., edge density). The GA has three parameters: the size p of the population, the percentage r of individuals to be crossed, and the percentage m of individuals to be mutated. The procedure is repeated (we set $P := P'$) until the fitness value of an individual in P is larger than a threshold θ. The algorithm therefore requires more extensive parameter tuning than SA. Moreover, the execution time and memory usage are higher (at each iteration we need to compute order of p interpretations).

V. Complexity

In this section we discuss the complexity of our WSD algorithm. Recall from Section III-B that the algorithm proceeds on a sentence-by-sentence basis and consists essentially of two steps, a graph construction phase and a disambiguation phase. Let k be a constant denoting the highest number of senses a word can have in WordNet:

$$k = \max_{w \in \text{WN}} |Senses(w)| \tag{13}$$

	Measure	Runtime
Local	Degree	$O(n)$
	PageRank	$O(n^2)$
	HITS	$O(n^2)$
	KPP	$O(nm + n^2 logn)$
	Betweenness	$O(nm)$
Global	Compactness	$O(\mu n^2)$
	Graph Entropy	$O(\mu n)$
	Edge Density	$O(\mu n)$

TABLE III
Time complexity of local and global measures.

Given a sentence σ with n words, the number of senses in the associated sentence graph V_σ is bound by:

$$|V_\sigma| = \Big| \bigcup_{i=1}^{n} Senses(w_i) \Big| \leq \sum_{i=1}^{n} |Senses(w_i)| \leq \sum_{i=1}^{n} k = kn \tag{14}$$

Our graph construction procedure executes a DFS from each vertex $v \in V_\sigma$. The running time for this is $O(|V_{\text{WN}}| + |E_{\text{WN}}|)$ [58], where V_{WN} and E_{WN} are the sets of vertices and edges in the entire WordNet graph. The number of edges $|E_{\text{WN}}|$ can be bound by $k'|V_{\text{WN}}|$, where k' is a constant equal to the maximum number of WordNet edges incident to any node in the WordNet graph.[4] So, the cost of a single DFS is $O(|V_{\text{WN}}| + k'|V_{\text{WN}}|) = O(|V_{\text{WN}}|)$. Given that the number of vertices in V_σ is bound by $kn \in O(n)$, the overall running time of the graph construction phase is $O(n|V_{\text{WN}}|)$. In practice, however, the running time is closer to $O(n^2)$. In fact, a DFS can take $O(n)$ time since we do not explore the entire WordNet network, the maximum number of incident edges is a small constant and the number of vertices visited during the search must be at distance $\leq L$.

With a local measure l, the running time of the disambiguation phase is $O(c_l(n))$, where $c_l(n)$ is the time cost incurred by l when applied to all $O(n)$ nodes in the sentence graph. We report running times for each measure individually in Table III. As can be seen, Degree complexity amounts to $O(n)$ time if the sentence graph is represented with an adjacency list. Measures of eigenvector centrality require the application of the power method, and thus take $O(n^2)$ time. The time complexity of KPP [59] depends on the calculation of all shortest paths which costs $O(nlogn + m)$ for each vertex; here, m is the number of edges in the sentence graph and $m \in O(n)$, as each vertex has a constant upper bound on the number of incident edges. Finally, for betweenness an $O(nm)$-implementation has been described [60].

Complexity increases substantially when global measures are employed. Calculating the score of a single graph (corresponding to one interpretation for sentence σ) takes $O(n^2)$ time ($O(n)$ for Graph Entropy and Edge Density if an adjacency list is used). Exhaustively generating all possible interpretations is computationally prohibitive with $O(k^n)$ complexity (recall that k is the maximum number of senses for a word in WordNet). Fortunately, we can reduce the search space to a very large constant μ using the approximation algorithms described in Section VI. The running time of the approximated global measures is thus polynomial.

[4] In WordNet 2.0 $k' = 627$, however only 80 nodes out of 115,510 have a degree greater than 100.

VI. Experimental Setup

A. Data

We evaluated our connectivity measures on three standard benchmark data sets. Specifically, we used the SemCor corpus [47] and the Senseval-3 [5] and Semeval-2007 [6] data sets. SemCor is the largest publicly available sense-tagged corpus. It is composed of 352 documents extracted from the Brown corpus. In 186 of these, all open class words (nouns, verbs, adjectives, adverbs) have been sense annotated. The remaining 166 have sense tags for verbs only. The corpus was created to provide examples of senses in context. The order of senses in WordNet is based on their SemCor frequency. SemCor has been used extensively in many WSD approaches, both supervised and unsupervised.

The Senseval-3 and Semeval-2007 data sets are subsets of the Wall Street Journal corpus. They each contain 2,037 and 465 words annotated with WordNet senses. They are publicly available and distributed as part of the Senseval (and Semeval) WSD evaluation exercises.[5] Our experiments were run in the all-words setting: the algorithm must disambiguate all (content) words in a given text (in contrast, in the lexical sample setting the data set consists of a selected few words and the system must disambiguate only these words).

The above data sets are tagged with different versions of WordNet. SemCor uses version 1.6, Senseval-3 version 1.7.1 and Semeval-2007 version 2.1. These were normalized to WordNet 2.0 using publicly available sense mappings.[6]

B. Reference Lexicons

An important prerequisite to the graph-based algorithm described in Section III-B is the reference lexicon that provides the meaning distinctions as well the lexical and semantic relations. Our experiments used the publicly available WordNet and an extended version created by the first author [61], which we refer to as EnWordNet. The latter contains additional edges (approximately 60,000) that link concepts across parts of speech via *collocational* relations. Such information is not explicitly encoded in WordNet, despite strong evidence that it helps in determining a word's sense in context [62], [63].

EnWordNet was created as follows. First, a list of collocations was compiled from the Oxford Collocations [64] and the Longman Language Activator [65] dictionaries as well as collocation web sites. These were word pairs consisting of a base w (e.g., drink_v) and its collocate w' (e.g., water_n), each of which had to be attested in WordNet. Next, each base w was disambiguated manually with an appropriate WordNet sense S. The set of collocates $\{w' : (S, w')\}$ was disambiguated automatically with the algorithm proposed in [15] while treating S as context. Disambiguated collocations were manually validated with the aid of a web interface. This process resulted in an enriched version of WordNet 2.0 which included the same synsets as the original with additional relation edges. Figure 3 shows an excerpt of this graph centered around drink_v^1. New nodes that have been added to WordNet are drawn as light gray ellipses (compare Figure 1).

To establish the degree to which WordNet and EnWordNet diverge, we examined their graph structure more closely [41]. Specifically, we analyzed how each dictionary fares in terms of the following properties:

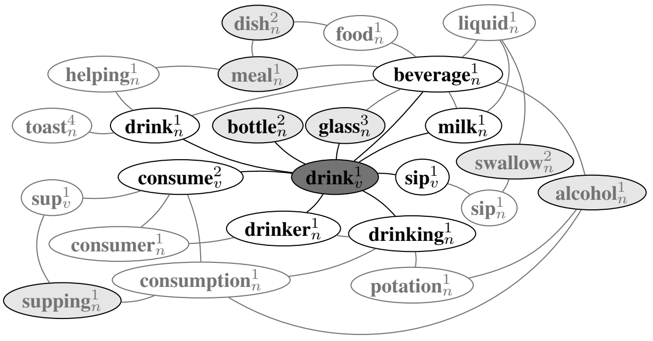


Fig. 3. An excerpt of the Enriched WordNet graph centered around drink_v^1.

1) *The small world effect*, which implies that most pairs of vertices are possibly connected by a short path through the network. The small world effect is determined as the mean shortest distance between vertex pairs in the graph:
$$l = \frac{1}{\frac{1}{2}n(n-1)} \sum_{i \geq j} d(i,j).$$
where $d(i,j)$ is the length of the shortest path between vertices i and j. A high value of l indicates a low small world effect, i.e., the degree of reachability between pairs of vertices in the graph is on average low. Following [41], we set $d(i,j)$ to zero if vertex j cannot be reached from i through an edge path in the graph (notice that this definition is different from the one given in equation 1).
2) *The clustering rate* (or *transitivity*), which quantifies the number of triangles in the graph — sets of three vertices each of which is connected to each of the others. In social networks terminology, the friend of your friend is likely to be your friend. The clustering rate is defined as:
$$C = \frac{3 \cdot \textit{number of triangles in the network}}{\textit{number of connected triples of vertices}},$$
where a "connected triple" is a single vertex with edges running to an unordered pair of others. A high C indicates that the associates of words are also directly associated. Clustering rate is complementary to the small world effect. The former measures graph connectivity in terms of interconnected neighborhoods, whereas the latter in terms of path length.
3) *The cumulative degree distribution*, that is the probability that the degree is greater than or equal to k:
$$P_k = \sum_{k'=k}^{\infty} p_{k'},$$
where $p_{k'}$ is the fraction of vertices in the graph that have degree k'. The cumulative distribution of complex networks such as the Web typically follows a power law.

Table IV shows the small world effect (l) and clustering rate (C) for the two dictionaries. EnWordNet has a considerably smaller mean shortest distance than WordNet (5.466 versus 7.243 edges) which indicates that it is more densely connected. Further evidence comes from the clustering rate which is increased in EnWordNet by a factor of 2. Figure 4 plots the cumulative degree distributions for both dictionaries in log-log coordinates. As can be seen, both distributions follow a power law, with the line for EnWordNet being slightly smoother due to the increased connectivity discussed above.

[5] For more information see `http://www.senseval.org/`.

[6] Available from `http://www.lsi.upc.edu/~nlp/web/`.

Graph	l	C
WordNet	7.243	0.047
EnWordNet	5.466	0.099

TABLE IV
NETWORK PROPERTIES WORDNET AND ENWORDNET.

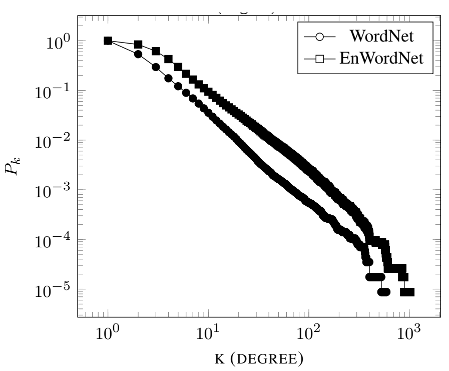


Fig. 4. Cumulative degree distribution for WordNet and EnWordNet.

C. Graph Construction and Search

In order to speed up the graph construction process, all paths connecting pairs of senses in both versions of WordNet were exhaustively enumerated and stored in a database which was consulted at run-time during disambiguation. We determined the optimal value of the maximum path length L experimentally. We ran our WSD algorithm on the SemCor data set using the Degree connectivity measure and the WordNet sense inventory while varying the path length from 0 to 10. We obtained 39% WSD accuracy at path length 0 (corresponding to random selection). This steadily increased until it reached a plateau at path length 4 with 50% accuracy. Longer paths did not bring additional gains. We thus fixed the maximum path length to 6, as it was longer than 4 but not as computationally expensive as 10.

When interfaced with global connectivity measures, our algorithm relies on a heuristic search procedure to find the best ranking interpretation (cf. Section IV-B). We compared the performance of SA and GAs for this task. Our WSD algorithm was run with the Edge Density measure and the WordNet sense inventory. The number of iterations u for SA was set to $5,000$ and, following [56], the constant T, initially set to 1.0, was reset to $T := 0.9 \cdot T$ after the u iterations. SA obtained an accuracy of 43.54%. As mentioned earlier (in Section IV-B) the GAs have more parameters. We performed a set of experiments with $p \in \{50, 100\}$, $r \in \{0.3, 0.6\}$, $m = 0.1$ and $\theta \in \{0.1, 0.2, 0.3\}$. The highest accuracy was 39.20% (with parameters $p = 50, r = 0.3, m = 0.1, \theta = 0.2$) which is more than 4% below the result obtained with SA. We therefore opted for SA in all subsequent experiments with global measures.

D. Evaluation

We evaluated our graph-based algorithm using the publicly available Senseval-3 scorer.[7] For any given system, the scorer reports precision (the number of correct senses over the number of senses returned), recall (the number of correct senses over the total number of senses identified in the evaluation data set), and their combined F1 measure ($\frac{2PR}{P+R}$). Since our method provides an answer for all ambiguous words, precision, recall, and F1 are the same.[8] We first examined the behaviour of the graph connectivity measures described in Section IV exhaustively on SemCor, the largest sense tagged data set available. Next, we evaluated the best performing measure on the smaller Senseval-3 and Semeval-2007 data sets and compared it with the state of the art.

We also compared our results against two baselines. The first one randomly selects a sense for an ambiguous word. The second baseline is Lesk's WSD algorithm, originally introduced in [16]. The algorithm assigns a sense to an ambiguous word by comparing the dictionary definitions (*glosses* in WordNet) of each of its senses with the words in the surrounding context. The sense whose definition has the highest overlap with the context is assumed to be the correct one. Following [45] we not only look for overlap between the glosses of two senses, but also between the glosses of their hypernyms, hyponyms, meronyms, holonyms, and troponyms. These *extended glosses* increase the algorithm's coverage. We also adopt their overlap scoring mechanism which treats each gloss as a bag of words and assigns an n word overlap the score of n^2. Analogously to our graph-based algorithm we disambiguate words on a sentence-by-sentence basis and break ties randomly. Our in-house implementation followed closely [45], the main difference being special purpose I/O routines to handle the relatedness relation from EnWordNet.

Finally, as an upper bound, we used the first-sense heuristic which assigns all instances of an ambiguous word its most frequent sense according to SemCor. It is important to note that current unsupervised WSD approaches—and also many supervised ones—rarely outperform this heuristic [10].

VII. RESULTS

A. Experiments on SemCor

Our results on SemCor are summarized in Table V. We report performance for WordNet and EnWordNet. The column "All" shows results on all words (monosemous and polysemous), whereas column "Poly" on the subset of polysemous words.

Let us first concentrate on the results we obtained with the standard WordNet inventory. As can be seen, all local measures perform better than the random sense baseline (Random) and worse than the first sense upper bound (First Sense)[9]. Degree, PageRank and Betweeness are significantly better than the extended Lesk algorithm (ExtLesk; $p < 0.01$ using a χ^2 test). KPP is significantly better than ExtLesk on polysemous words (see the column Poly in Table V) but the advantage disappears when taking all words into account. HITS performs significantly worse than ExtLesk, Degree and PageRank significantly outperform all other local measures ($p < 0.01$). In fact, the two measures yield similar results (their difference is not statistically significant). This is not entirely surprising; the PageRank value of a node is proportional to its degree in undirected graphs. Previous research on directed graphs has also experimentally shown that the two measures are broadly equivalent [66]. Among the global measures, Compactness and Edge Density are significantly better than Graph Entropy ($p < 0.01$). However, when compared to local measures and the ExtLesk, global measures lag behind

[7] www.cse.unt.edu/~rada/senseval/senseval3/scoring/

[8] F1 ranges from 0 to 1. We report percentages in Tables V, VII, and VIII for the sake of readability.

[9] The First Sense performance is an upper bound on SemCor as it is determined on the sense-tagged version of the very same corpus.

		WordNet		EnWordNet	
	Measure	All	Poly	All	Poly
	Random	39.13	23.42	39.13	23.42
	ExtLesk	47.85	34.05	48.75	35.25
Local	**Degree**	**50.01**	**37.80**	**56.62**	**46.03**
	PageRank	49.76	37.49	56.46	45.83
	HITS	44.29	30.69	52.40	40.78
	KPP	47.89	35.16	55.65	44.82
	Betweenness	48.72	36.20	56.48	45.85
Global	Compactness	43.53	29.74	48.31	35.68
	Graph Entropy	42.98	29.06	43.06	29.16
	Edge Density	43.54	29.76	52.16	40.48
	First Sense	74.17	68.80	74.17	68.80

TABLE V

PERFORMANCE OF CONNECTIVITY MEASURES ON SEMCOR.

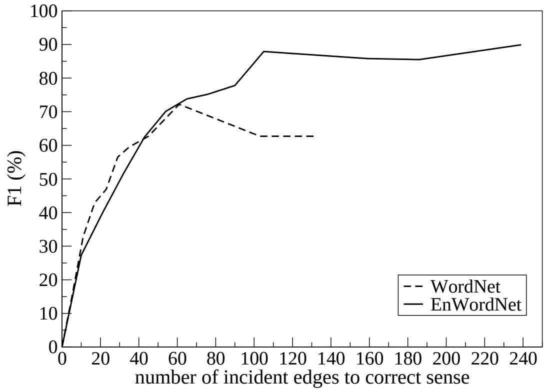


Fig. 5. Performance of Degree by number of incident edges for polysemous words in SemCor.

(all differences are statistically significant). We conjecture that the inferior performance of the global measures is due to the necessary heuristic algorithm used for searching the interpretation space (see Section VI-C).

We now discuss the performance of the different measures under the enriched WordNet. Here, we also observe that all measures are significantly worse than the first sense but better than the random baseline. The best global measure is Edge Density, which significantly outperforms Compactness and Graph entropy and ExtLesk ($p < 0.01$). The best local measures are Degree, Betweenness and PageRank (their difference is not statistically significant). They all are significantly better than Edge Density and ExtLesk.

A denser reference lexicon with a large number of semantic relations seems to benefit both local and global measures. The enriched WordNet yields better results in comparison to vanilla WordNet across the board. Using the former sense inventory increases disambiguation performance in most cases by approximately 10% (see the column Poly in Table V). The benefit of EnWordNet is perhaps most apparent with Edge Density which becomes competitive with HITS despite the combinatorial explosion problem discussed above. This makes intuitively sense. The denser the sense inventory is, the more appropriate edge-based measures will be at capturing sense distinctions. Note that Graph Entropy seems insensitive to the use of a denser sense inventory. The differences between WordNet and EnWordNet are not radical enough to result in changes in graph entropy. For this reason, similar performances are observed with both dictionaries.

Lexicon	H	N	S	M	A	C
WordNet	9.29	1.21	0.33	0.29	0.11	–
EnWordNet	12.80	1.88	0.80	0.41	0.18	20.50

TABLE VI

AVERAGE NUMBER OF OUTGOING EDGES BY KIND FOR THE SENSES SELECTED BY DEGREE WITH WORDNET AND ENWORDNET.

We further analyzed the influence of the sense inventory on the WSD algorithm by examining how Degree's performance varies on SemCor as the number of incident edges of the correct target word sense increases. Figure 5 shows that F1 tends to increase when the target sense has a large number of incident edges. This is especially the case for EnWordNet. F1 starts dropping for WordNet when a degree of 60 is reached. This is probably due to the fact that high-degree vertices in WordNet typically have a large number of hyponyms (e.g., $period_n$, $dish_n$). This often degrades performance, as too many specializations may lead to accidental sense matches. In contrast, a vertex in EnWordNet with a high degree is related to many other vertices transversally (i.e., in a non-taxonomic manner). Thus, the chance of identifying semantically valid paths is increased. In fact, when the correct sense of a target word is adjacent to more than 100 edges (this holds for almost 20,000 test items in SemCor), Degree obtains a precision of 88% with EnWordNet (and only 62% with WordNet).

To get more insight into the kinds of semantic relations that contribute to better performance we estimated their average distribution over the set of outgoing edges for each sense chosen by Degree in SemCor. Table VI reports results for hypernymy/hyponymy (H), nominalization (N), similarity (S), meronymy/holonymy (M), antonymy (A), and collocation-based relations (C). The latter refer to relations present in EnWordNet but not in WordNet. We only consider relations which constitute at least 0.10 of the incident edges per sense (on average). We observe that the contribution of each relation significantly increases from WordNet to EnWordNet. Expectedly, hypernymy/hyponymy is an important relation in both dictionaries. Collocation-based relations have a prominent role in EnWordNet, thus confirming our hypothesis that syntagmatic relations play a key role in WSD performance.

In the results reported thus far ties are broken randomly, as we did not want to make use of any knowledge sources other than the WordNet graph. However, it is customary in the WSD literature to resolve ties by choosing the most frequent sense in SemCor. To establish how much could be gained from this strategy, we modified our algorithm so that it defaults to the first WordNet sense in case of ties. Degree obtained an F1 of 53.08% on SemCor. F1 increased to 59.83% with EnWordNet. We observed similar gains (i.e., within the range of 3%) for the other connectivity measures.

In sum, our results indicate that Degree is the best connectivity measure overall as it performs consistently well across sense inventories. This is an encouraging result: Degree is one of the simplest connectivity measures available, with moderate running time complexity. Indeed, we find that more sophisticated measures such as HITS perform comparably to the extended Lesk algorithm. The latter does not take advantage of WordNet's graph structure in any way (it selects the sense whose definition has the highest overlap with the surrounding context words). We also find that the lexicon at hand plays an important role in graph-based WSD. The results in Table V suggest that advances in

System	F1
Best Unsupervised (Sussex)	45.8
ExtLesk	43.1
Degree Unsupervised	52.9
Best Semi-supervised (IRST-DDD)	56.7
Degree Semi-Unsupervised	60.7
First Sense	62.4
Best Supervised (GAMBL)	65.2

TABLE VII
RESULTS ON THE SENSEVAL-3 ALL-WORDS DATA SET. DEGREE IS COMPARED TO LESK AND THE BEST UNSUPERVISED, SEMI-SUPERVISED, AND SUPERVISED SYSTEMS IN THE SENSEVAL-3 COMPETITION.

System	F1
ExtLesk	31.9
Best Unsupervised (JU-SKNSB)	40.2
Degree Unsupervised	43.1
First Sense	51.4
Best Semi-supervised (RACAI)	52.7
Degree Semi-Supervised	53.0
Best supervised (PN-NL)	59.1

TABLE VIII
RESULTS ON THE SEMEVAL ALL-WORDS DATA SET. DEGREE IS COMPARED TO LESK AND THE BEST UNSUPERVISED, SEMI-SUPERVISED, AND SUPERVISED SYSTEMS THAT ENTERED THE SEMEVAL COMPETITION.

WSD performance can be made simply by creating better sense connections with a larger number of collocational relations.

B. Experiments on Senseval-3 and Semeval-2007 All Words

In this section we situate the WSD algorithm proposed in this paper with regard to the state of the art. Broadly speaking, WSD algorithms come in three flavors. Supervised systems learn disambiguation cues from hand-labeled data and usually demonstrate superior performance to other methods. Unsupervised systems, like the one presented here, do not utilize sense tagged data in any way, and semi-supervised approaches use some annotated data selectively, e.g., by defaulting to the SemCor first sense when the system is not confident about the correctness of its answers.

Table VII shows the performance of our algorithm on the Senseval-3 all-words data set using Degree and the EnWordNet. The algorithm is run in an unsupervised setting without making use of the first sense and a semi-supervised setting where it backs off to the first sense when the Degree score for the target sense is below a certain (empirically estimated) threshold. The latter was set to 0.11 after maximizing Degree's F1 on 100 randomly chosen word instances from the Senseval-2 all-words test set [67]. Degree is also compared to Lesk, the first sense heuristic, and the best supervised, unsupervised and semi-supervised systems that participated in Senseval-3.

As can be seen in Table VII, Degree in the unsupervised setting outperforms Lesk and the best unsupervised Senseval-3 system (Sussex). Both differences are statistically significant ($p < 0.01$) using bootstrap resampling. The latter system [10] implements a method that does not rely on contextual cues for disambiguation. Instead it acquires first senses automatically by quantifying the degree of similarity between the distributional neighbors and the sense descriptions of an ambiguous word. Degree has an advantage over [10] as it disambiguates words in context and is not restricted to a unique first sense.

In the semi-supervised setting, Degree is not significantly worse than the first sense heuristic, but is significantly better ($p < 0.01$) than the best semi-supervised Senseval-3 system (IRST-DDD) which implements a similarity-based algorithm. The system was developed by [68] and performs domain driven disambiguation. It compares the domain of the context surrounding the target word with the domains of its senses using a version of WordNet augmented with domain labels (e.g., ECONOMY, GEOGRAPHY, etc.). Finally, the best supervised system (GAMBL) is significantly better than Degree in both the unsupervised and semi-supervised settings (see Table VII). This is not surprising: GAMBL [69] is a memory-based system which learns word experts from an extensive collection of training sets. In contrast, Degree has access to no (unsupervised) or very little (semi-supervised) training data. Also note that GAMBL (significantly) outperforms the first sense heuristic, albeit by a small margin (2.8%).

We replicated the experiment just described on the most recent all-words WSD data set, which was created in the context of the Semeval-2007 competition [70]. Again we compared Degree in the two settings (unsupervised and semi-supervised) against Lesk and the best systems that participated in Semeval-2007. Although the Semeval test set is generally more difficult than Senseval-3 — it has a higher number of verbs which are notoriously hard to disambiguate — we observe similar trends. Degree in the unsupervised setting is significantly better than Lesk. It is also numerically better than the best unsupervised system (JU-SKNSB) which is a modified version of the extended Lesk algorithm [71]. However, the difference is not statistically significant (the Semeval-2007 test set is relatively small, containing only 465 instances). The semi-supervised Degree, the best semi-supervised Semeval system (RACAI), and the first sense heuristic are all in the same ballpark (the differences are not statistically significant). RACAI [72] is inspired by lexical attraction models [73]; it finds the combination of senses that is most likely in a sentence represented by dependency-like structures. Expectedly, Degree (and all other unsupervised and semi-supervised approaches) are significantly worse than the best supervised system (PN-NL) which is essentially a maximum entropy classifier with a rich feature set [74].

Overall, we find that our graph-based algorithm yields competitive performance with the state of the art. Bear in mind that our approach is relatively simple and parameter free. All we need is a lexicon and a means of assessing the importance of graph nodes. We do not use any sense annotated data for training or any syntactic information other than part of speech tagging. It is therefore interesting to see that our algorithm fares well in comparison with other more sophisticated methods that exploit syntactic information (e.g., [10], [72]) or additional resources (e.g., [68]).

VIII. CONCLUSIONS

In this paper we presented a study of graph connectivity measures for unsupervised WSD. We evaluated a wide range of local and global measures with the aim of isolating those that are particularly suited for this task. Our results indicate that local measures yield better performance than global ones. The best local measures are Degree and PageRank. A similar conclusion is drawn in [66] in the context of Web page ranking. Furthermore, [75] prove that the two measures are closely related, obeying a similar power law distribution.

We also find that the employed reference dictionary critically influences WSD performance. We obtain a large improvement (in the range of 10%) when adopting a version of WordNet enriched with thousands of relatedness edges. This indicates that graph-based WSD algorithms will perform better with more densely connected sense inventories, with more incident edges for every node. An interesting future direction would be to investigate ways of automatically enhancing WordNet (and similar dictionaries) with relatedness information. For example, by adding edges in the graph for nodes whose distributional similarity exceeds a certain threshold.

Beyond the specific algorithm presented in this paper, our results are relevant for other graph-based approaches to word sense disambiguation [9], [15] and discrimination [29], [30]. Our experiments show that performance could potentially increase when the right connectivity measure is chosen. The proposed measures are independent of the adopted reference lexicon; they induce a sense ranking solely by considering graph connectivity and can thus be ported across algorithms, languages, and sense inventories. WordNet-like reference lexicons exist for several languages (e.g., EuroWordNet [76], MultiWordNet [77]). Methods also have been developed for learning taxonomies from machine readable dictionaries [78], [79] and corpora [80]. It is an interesting future direction to establish how well our WSD method performs with such resources.

Our experiments focused primarily on graph connectivity measures and their suitability for WordNet-like sense inventories. For this reason, we employed a relatively generic WSD algorithm (see Section III) without extensive tuning and obtained state-of-the-art performance when assessing our system on standard evaluation data sets (e.g., Senseval-3 and Semeval-2007). However, this does not mean to say that the algorithm could not be further improved. For instance, we could consider word sequences larger than sentences, take into account syntactic relations, or score edges in the graph according to semantic importance (e.g., hypernymy is more important than meronymy).

More research is needed to assess whether our results extend to other NLP tasks, besides WSD. An obvious application would be summarization, where graph-based methods have met with reasonable success [33], [34]. Eigenvector centrality measures are a popular choice here; however, their performance against other graph connectivity measures has not yet been studied in detail.

Acknowledgments

We are grateful to the anonymous referees whose feedback helped to substantially improve the present paper. This work was supported by EPSRC (grant EP/C538447/1) and the Royal Society of Edinburgh. A preliminary version was published in the proceedings of IJCAI 2007.

References

[1] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller, "Word-sense disambiguation for machine translation," in *Proc. of HLT and EMNLP*, Vancouver, BC, Canada, 2005, pp. 771–778.

[2] C. Stokoe, "Differentiating homonymy and polysemy in information retrieval," in *Proc. of HLT and EMNLP*, Vancouver, BC, Canada, 2005, pp. 403–410.

[3] G. Ramakrishnan, A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya, "Question answering via Bayesian inference on lexical relations," in *Proc. of the ACL Workshop on Multilingual Summarization and Question Answering*, Sapporo, Japan, 2003, pp. 1–10.

[4] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in *Proc. of the ACL Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 1997, pp. 10–17.

[5] B. Snyder and M. Palmer, "The English all-words task," in *Proc. of Senseval-3 Workshop*, Barcelona, Spain, 2004, pp. 41–43.

[6] S. Pradhan, E. Loper, D. Dligach, and M. Palmer, "Semeval-2007 task-17: English lexical sample, SRL and all words," in *Proc. of Semeval-2007 Workshop*, Prague, Czech Republic, 2007, pp. 87–92.

[7] D. Yarowsky and R. Florian, "Evaluating sense disambiguation across diverse parameter spaces," *Natural Language Engineering*, vol. 9, no. 4, pp. 293–310, 2002.

[8] T. H. Ng, "Getting serious about word sense disambiguation," in *Proc. of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, DC, 1997, pp. 1–7.

[9] R. Mihalcea, "Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling," in *Proc. of HLT/EMNLP*, Vancouver, BC, Canada, 2005, pp. 411–418.

[10] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll, "Finding predominant senses in untagged text," in *Proc. of the 42nd ACL*, Barcelona, Spain, 2004, pp. 280–287.

[11] H. Schütze, "Automatic word sense discrimination," *Computational Linguistics*, vol. 24, no. 1, pp. 97–124, 1998.

[12] D. Lin and P. Pantel, "Discovering word senses from text," in *Proceedings of the* 8^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002, pp. 613–619.

[13] E. Agirre and P. Edmonds, Eds., *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2007.

[14] R. Navigli, "Word sense disambiguation: a survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, 2009.

[15] R. Navigli and P. Velardi, "Structural semantic interconnections: a knowledge-based approach to word sense disambiguation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1075–1088, 2005.

[16] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proc. of the 5th SIGDOC*, New York, NY, 1986, pp. 24–26.

[17] S. Brody, R. Navigli, and M. Lapata, "Ensemble methods for unsupervised WSD," in *Proc. of the COLING/ACL*, Sydney, Australia, 2006, pp. 97–104.

[18] R. A. Botafogo, E. Rivlin, and B. Shneiderman, "Structural analysis of hypertexts: Identifying hierarchies and useful metrics," *ACM Transactions on Information Systems*, vol. 10, no. 2, pp. 142–180, 1992.

[19] P. Hage and F. Harary, "Eccentricity and centrality in networks," *Social Networks*, vol. 13, pp. 57–63, 1995.

[20] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge, UK: Cambridge University Press, 1994.

[21] S. Brin and M. Page, "Anatomy of a large-scale hypertextual Web search engine," in *Proc. of the 7th Conference on World Wide Web*, Brisbane, Australia, 1998, pp. 107–117.

[22] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proc. of the 9th SODA*, San Francisco, CA, 1998, pp. 668–677.

[23] Z. Gyngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proc. of the 30th VLDB*, Toronto, Canada, 2004, pp. 271–279.

[24] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: A new approach to topic-specific Web resource discovery," in *Proc. of the 8th Int'l Conference on World Wide Web*, NYC, 1999, pp. 1623–1640.

[25] A. Balmin, V. Hristidis, and Y. Papakonstantinou, "Objectrank: Authority-based keyword search in databases," in *Proc. of the 13th Conference on Very Large Data Bases*, Toronto, Canada, 2004.

[26] M. A. R. Johan Bollen and H. V. de Sompel, "Mesur: Usage-based metrics of scholarly impact," in *Proc. of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, Vancouver, BC, Canada, 2007, p. 474.

[27] M. Galley and K. McKeown, "Improving word sense disambiguation in lexical chaining," in *Proc. of the 18th IJCAI*, Acapulco, Mexico, 2003, pp. 1486–1488.

[28] R. Mihalcea, P. Tarau, and E. Figa, "Pagerank on semantic networks, with application to word sense disambiguation," in *Proc. of the 20th COLING*, Geneva, Switzerland, 2004.

[29] J. Véronis, "Hyperlex: lexical cartography for information retrieval," *Computer, Speech and Language*, vol. 18, no. 3, pp. 223–252, 2004.

[30] E. Agirre, D. Martínez, O. López de Lacalle, and A. Soroa, "Two graph-based algorithms for state-of-the-art WSD," in *Proc. of the EMNLP*, Sydney, Australia, 2006, pp. 585–593.

[31] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," CMU-CALD-02, Tech. Rep., 2002.

[32] Z. Niu, D. Ji, and C. L. Tan, "Word sense disambiguation using label propagation based semi-supervised learning," in *Proc. of 43rd ACL*, Ann Arbor, MI, 2005, pp. 395–402.

[33] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[34] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proc. of EMNLP*, D. Lin and D. Wu, Eds., Barcelona, Spain, 2004, pp. 404–411.

[35] X. Wan and J. Yang, "Improved affinity graph based multi-document summarization," in *Proc. of the HLT/NAACL*, New York City, NYC, 2006, pp. 181–184.

[36] A. Esuli and F. Sebastiani, "Pageranking WordNet synsets: An application to opinion mining," in *Proc. of the 45th ACL*, Prague, Czech Republic, June 2007, pp. 424–431.

[37] J. Otterbacher, G. Erkan, and D. Radev, "Using random walks for question-focused sentence retrieval," in *Proc. of HLT and EMNLP*, Vancouver, BC, Canada, 2005, pp. 915–922.

[38] R. Navigli and P. Velardi, "Learning domain ontologies from document warehouses and dedicated websites," *Computational Linguistics*, vol. 30, no. 2, pp. 151–179, 2004.

[39] D. Feng, E. Shaw, J. Kim, and E. Hovy, "Learning to detect conversation focus of threaded discussions," in *Proc. of the HLT/NAACL*, New York City, NYC, 2006, pp. 208–215.

[40] K. Toutanova, C. Manning, and A. Ng, "Learning random walk models for inducing word dependency distributions," in *Proc. of 21st ICML*, Banff, Canada, 2004.

[41] M. E. J. Newman, "The structure and function of complex networks," *The SIAM Review*, vol. 45, pp. 167–256, 2003.

[42] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of semantic distance," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[43] T. Pedersen, S. Banerjee, and S. Patwardhan, "Maximizing semantic relatedness to perform word sense disambiguation," Minnesota, Tech. Rep. UMSI 2005/25, 2005.

[44] R. Navigli and M. Lapata, "Graph connectivity measures for unsupervised word sense disambiguation," in *Proceedings of the 20th IJCAI*, Hyderabad, India, 2007, pp. 1683–1688.

[45] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Proc. of the 18th IJCAI*, Acapulco, Mexico, 2003, pp. 805–810.

[46] C. Fellbaum, Ed., *WordNet: an Electronic Lexical Database*. MIT Press, 1998.

[47] G. Miller, C. Leacock, T. Randee, and R. Bunker, "A semantic concordance," in *Proc. of the 3rd DARPA Workshop on HLT*, Plainsboro, New Jersey, 1993, pp. 303–308.

[48] A. Novischi, "Combining methods for word sense disambiguation of WordNet glosses," in *Proc. of the 17th FLAIRS*, Miami Beach, FL, 2004.

[49] E. Agirre and A. Soroa, "Using the multilingual central repository for graph-based word sense disambiguation," in *Proc. of the 6th Language Resources and Evaluation Conference*, Marrakech, Morocco, 2008.

[50] W. Gale, K. Church, and D. Yarowsky, "One sense per discourse," in *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, Pacific Grove, CA, 1992, pp. 233–237.

[51] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, pp. 215–239, 1979.

[52] B. P. Bonacich, "Factoring and weighing approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, pp. 113–120, 1972.

[53] S. P. Borgatti, "Identifying sets of key players in a network," in *Proc. of the Conference on Integration of Knowledge Intensive Multi-Agent Systems*, Boston, MA, 2003, pp. 127–131.

[54] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, pp. 581–603, 1966.

[55] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. USA: Prentice Hall; 2nd edition, 2002.

[56] J. Cowie, J. Guthrie, and L. Guthrie, "Lexical disambiguation using simulated annealing," in *Proc. of the 14th COLING*, Nantes, France, 1992, pp. 359–365.

[57] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[58] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*, ser. the MIT Electrical Engineering and Computer Science Series. Cambridge, MA: MIT Press, 1990.

[59] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal for the Association of Computing Machinery*, vol. 24, pp. 1–13, 1977.

[60] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[61] R. Navigli, "Semi-automatic extension of large-scale linguistic knowledge bases," in *Proc. of the 18th FLAIRS*, Clearwater Beach, Florida, 2005, pp. 548–553.

[62] D. Yarowsky, "One sense per collocation," in *Proc. of the ARPA Workshop on HLT*, Princeton, NJ, 1993, pp. 266–271.

[63] H. T. Ng and H. B. Lee, "Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach," in *Proc. of the 34th ACL*, Santa Cruz, CA, 1996, pp. 40–47.

[64] D. Lea, Ed., *Oxford Collocations*. Oxford University Press, 2002.

[65] Longman, Ed., *Longman Language Activator*. Pearson Education, 2003.

[66] T. Upstill, N. Craswell, and D. Hawking, "Predicting fame and fortune: PageRank or Indegree?" in *Proc. of the 8th Australasian Document Computing Symposium*, Canberra, Australia, 2003, pp. 31–40.

[67] P. Edmonds and S. Cotton, "Senseval-2: Overview," in *Proc. of Senseval-2 Workshop*, Toulouse, France, 2001, pp. 1–6.

[68] C. Strapparava, A. Gliozzo, and C. Giuliano, "Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3," in *Proc. of Senseval-3 Workshop*, Barcelona, Spain, 2004, pp. 229–234.

[69] B. Decadt, V. Hoste, W. Daelemans, and A. van den Bosch, "GAMBL, genetic algorithm optimization of memory-based WSD," in *Proc. of Senseval-3 Workshop*, Barcelona, Spain, 2004, pp. 108–112.

[70] E. Agirre, L. Màrquez, and R. Wicentowski, Eds., *Proc. of Semeval-2007 Workshop*. Prague, Czech Republic: Association for Computational Linguistics, 2007.

[71] S. K. Naskar and S. Bandyopadhyay, "Ju-sknsb: Extended WordNet based WSD on the english all-words task at semeval-1," in *Proc. of Semeval-2007 Workshop*, Prague, Czech Republic, 2007, pp. 203–206.

[72] R. Ion and D. Tufiş, "RACAI: Meaning affinity models," in *Proc. of Semeval-2007 Workshop*, Prague, Czech Republic, 2007, pp. 282–287.

[73] D. Yuret, "Discovery of linguistic relations using lexical attraction," Ph.D. dissertation, Department of Computer Science and Electrical Engineering, MIT, 1998.

[74] S. Tratz, A. Sanfilippo, M. Gregory, A. Chappell, C. Posse, and P. Whitney, "PNNL: A supervised maximum entropy approach to word sense disambiguation," in *Proc. of Semeval-2007 Workshop*, Prague, Czech Republic, June 2007, pp. 264–267.

[75] N. Litvak, , W. Scheinhardt, and Y. Volkovich, "In-degree and PageRank of Web pages: Why do they follow similar power laws?" *Memorandum 1807 Department of Applied Mathematics, University of Twente, Enschede. ISSN 0169-2690*, 2006.

[76] P. Vossen, Ed., *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht, The Netherlands: Kluwer, 1998.

[77] E. Pianta, L. Bentivogli, and C. Girardi, "MultiWordNet: Developing an aligned multilingual database," in *Proc. of the 1st International Conference on Global WordNet*, Mysore, India, 2002, pp. 21–25.

[78] M. Chodorow, R. Byrd, and G. Heidorn, "Extracting semantic hierarchies from a large on-line dictionary," in *Proc. of the 23rd ACL*, Chicago, IL, 1985, pp. 299–304.

[79] G. Rigau, H. Rodríguez, and E. Agirre, "Building accurate semantic taxonomies from monolingual mrds," in *Proc. of the 17th COLING*. Association for Computational Linguistics, 1998, pp. 1103–1109.

[80] R. Snow, D. Jurafsky, and A. Y. Ng, "Semantic taxonomy induction from heterogenous evidence," in *Proc. of the COLING/ACL*, Sydney, Australia, 2006, pp. 801–808.

Roberto Navigli is an Assistant Professor in the Department of Computer Science at the University of Rome "La Sapienza". He received a PhD in Computer Science from "La Sapienza", and was awarded with the Marco Cadoli 2007 Italian national prize for the best PhD thesis in Artificial Intelligence. His research focuses on natural language processing (specifically, word sense disambiguation, knowledge acquisition, ontology learning, semantic social network analysis and the semantic Web).

Mirella Lapata received a Bsc (Hons) degree in computer science from the University of Athens in 1994 and an Msc degree from Carnegie Mellon University in 1998. She received a PhD degree in Natural Language Processing from the University of Edinburgh in 2001. She is a Reader in Edinburgh's School of Informatics. Her research focuses on probabilistic learning techniques for natural language understanding and generation. Examples include word sense disambiguation, discourse modeling, lexicon acquisition, and document summarization.