Word Sense Disambiguation and Semantics techniques

Amaal Saleh Hasan
Sultan Qaboos University, College Of Science, Computer Science Department
Oman,
amaalh@squ.edu.om

Abstract

Of the many kinds of ambiguity in language is word sense. This problem had received the most attention in computational linguistics, and the reasons for this are clear: their resolution is seemingly essential for any practical application, and they seem to require a wide variety of methods and knowledge-sources with no pattern apparent in what any particular instance requires.

Since the beginning of NLP field and the researchers trying to bypass the bottleneck of word sense disambiguation (WSD). The lack of high-performing methods for sense disambiguation may be considered the major obstacle that prevented an extensive use of natural language processing techniques in many areas of information technology, such as information classification and retrieval, query processing, advanced Web search, document warehousing, etc.

This obstacle inspired many researchers to come up with new methods, approaches, and to exploit whatever knowledge resources exist to develop a solution. Previously most of the techniques used were based on a huge amount of knowledge sources, which suffers from the problems of incomplete, inadequate, and expensive resources.

After that a trained techniques and the statistical approaches dominant over the field for a while were at the beginning good results obtained with limited and small domains. But with the expanding of the domains, the luck of good trained knowledge sources besides the neglecting of the semantic roles did not lay good results at the end.

Now a day’s the research had been directed towards a hybrid approach that depends on the available and more reliable knowledge sources besides the semantic techniques, and training methods which consider the role of the words in its context.

In this research we will take a case study of WSD. Then we will display the semantic techniques used in it and display the effect of implementing consistence concept evaluation method to produce more reliable results based on semantic similarity, besides that we will show how more conceptualization of the semantic relations can be implemented through semantic information.

Key words: WSD, semantic similarity, structural semantic interconnection SSI.

1. Introduction

One of the first problems that are encountered by any natural language processing system is that of lexical ambiguity, be it syntactic or semantic. The resolution of a word's syntactic ambiguity has largely been solved in language processing by part-of-speech taggers which predict the syntactic category of words in text with high levels of accuracy. The problem of resolving semantic ambiguity is generally known as word sense disambiguation and has proved to be more difficult than syntactic disambiguation.

The problem is that words often have more than one meaning, sometimes fairly similar and sometimes completely different. The meaning of a word in a particular usage can only be determined by examining its context. For example consider the following two sentences, each with a different sense of the word *bank* :

- The boy leapt from the bank into the cold water.
- The van pulled up outside the bank and three masked men got out.

We immediately recognize that in the first sentence *bank* refers to the edge of a river and in the second to a building.
The problem of doing WSD by computer is not new; it goes back to the early days of machine translation.
But like other areas of computational linguistics, research into WSD has seen resurgence because of the availability of large corpora. Statistical methods for WSD, especially techniques in machine learning, have proved to be very effective, as SENSEVAL stated.

The SensEval workshop series are specifically dedicated to the evaluation of WSD algorithms. Systems compete on different tasks (e.g., full WSD on generic texts, disambiguation of dictionary sense definitions, automatic labeling of semantic roles) and in different languages. At Senseval-3, held in March 2004, 17 supervised and 9 unsupervised systems participated in the task. The best systems were those using a combination of several machine learning methods, trained with data on word co occurrences and, in few cases, with syntactic features, but nearly no system used semantic information.The best systems reached about 65 percent precision, 65 percent recall, a performance considered well below the needs of many real-world applications [6].

1.1 Definition of the problem
Word Sense Disambiguation (WSD) refers to a task that automatically assigns a sense, selected from a set of pre-defined word senses to an instance of a polysemous word in a particular context. It is an important but challenging technique and necessary for many real world applications such as machine translation (MT), semantic mapping (SM), semantic annotation (SA), and ontology learning (OL). It is also believed to be helpful in improving the performance of many applications such as information retrieval (IR), information extraction (IE), and speech recognition (SR).
WSD is difficult, because it involves much world knowledge or common sense, which is difficult to build or organize. Dictionaries provide the definition and partial lexical knowledge for each sense. But they include little well-defined world knowledge (or common sense). An alternative is for a program to automatically learn world knowledge from manually sense-tagged examples, called a training corpus.The conceptual model for WSD is shown in figure 1.

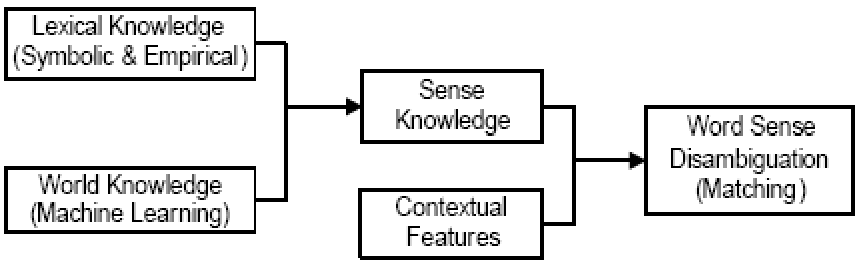

Figure 1 conceptual model of WSD

The problem of word sense disambiguation has been described as *AI-complete*, that is, a problem which can be solved only by first resolving all the difficult problems in artificial intelligence (AI), such as the representation of common sense and encyclopedic knowledge. However, now a days considerable progress was being made in the area of knowledge representation, especially the emergence of semantic networks, which were immediately applied to sense disambiguation.[7]
Many efforts had been dedicated for the creation of large lexical knowledge bases and annotated resources which in turn offer an ideal starting point for constructing structured representations of word senses. In these repositories, lexical knowledge is described with a variable degree of formality and many criticisms of the consistency and soundness (with reference to computer science standards) of the encoded information have been made. Despite these criticisms, these knowledge repositories became highly popular to the point where dedicated conferences are organized each year among the scientists that use these resources for a variety of applications in the information technology area like the SensEval workshop series mentioned above. [6]

2. Knowledge sources and algorithms

WSD heavily relies on knowledge. In fact, the skeletal procedure of any WSD system can be summarized as follows: given a set of words (e.g., a sentence or a bag of words), a technique is applied which makes use of one or more sources of knowledge to associate the most appropriate senses using words in context. Knowledge sources can vary considerably from corpora (i.e., collections) of texts, either unlabeled or annotated with word senses, to more structured resources, such as machine readable dictionaries, semantic networks, etc. Without knowledge, it would be impossible for both humans and machines to identify the meaning.
Unfortunately, the manual creation of knowledge resources is an expensive and time consuming effort, which must be repeated every time the disambiguation scenario changes (like in the presence of new domains, different languages, and even sense inventories). This is a fundamental problem which pervades the field of WSD, and is called the *knowledge acquisition bottleneck.*
So knowledge sources used for WSD are either lexical knowledge released to the public, or world knowledge learned from a training corpus.

2.1 Main lexical knowledge

WordNet [4] is a computational lexicon of English based on psycholinguistic principles, created and maintained at Princeton University. It encodes concepts in terms of sets of synonyms (called *synsets*). Version 3.0, contains about 155,000 words organized in over 117,000 synsets. For example, the concept of *automobile* is expressed with the following synset:

$$\{car_n^1, auto_n^1, automobile_n^1, machine_n^4, motorcar_n^1\}.$$

We can view a synset as a set of word senses all expressing the same meaning. The following function associates with each part-of-speech tagged word *wp* the set of its WordNet senses:

$$Senses_{WN} : \mathcal{L} \times \text{ POS} \rightarrow 2^{\text{SYNSETS}},$$

where SYNSETS is the entire set of synsets in WordNet. For example:

$$\begin{aligned}Senses_{WN}(car_n) = \{&\{car^1_n, auto^1_n, automobile^1_n, machine^4_n, motorcar^1_n\},\\ &\{car^2_n, rail\ car^1_n, rail\ way\ car^1_n, rail\ road\ car^1_n\},\\ &\{cable\ car^1_n, car^3_n\},\\ &\{car^4_n, gondola^3_n\},\\ &\{car^5_n, elevator\ car^1_n\}\}.\end{aligned}$$

For each synset, WordNet provides the following information:

1. A *gloss*, that is, a textual definition of the synset possibly with a set of usage examples (e.g., the gloss of *car*1 *n* is "a 4-wheeled motor vehicle; usually propelled by an internal combustion engine; 'he needs a car to get to work' ").
2. *Lexical and semantic relations*, which connect pairs of word senses and synsets, respectively: while semantic relations apply to synsets in their entirety (i.e., to all members of a synset), lexical relations connect word senses included in the respective synsets.
3. *Antonymy*: *X* is an antonym of *Y* if it expresses the opposite concept (e.g., *good*1*a* is the antonym of *bad*1*a*). Antonymy holds for all parts of speech.
4. *Hypernymy* (also called *kind-of* or *is-a*): *Y* is a hypernym of *X* if every *X* is a (kind of) *Y* (*motor vehicle*1 *n* is a hypernym of *car*1 *n*). Hypernymy holds between pairs of nominal or verbal synsets.
5. *Hyponymy* : the inverse relations of hypernymy for nominal and verbal synsets, respectively.
6. *Meronymy* (also called *part-of*): *Y* is a meronym of *X* if *Y* is a part of *X* (e.g., *flesh*3 *n* is a meronym of *fruit*1 *n*). Meronymy holds for nominal synsets only.
7. *Holonymy*: *Y* is a holonym of *X* if *X* is a part of *Y* (the inverse of meronymy).
8. *Entailment*: a verb Y is entailed by a verb X if by doing X you must be doing *Y* (e.g., *snore*1*v* entails *sleep*1*v*).
9. *Similarity*: an adjective *X* is similar to an adjective *Y* (e.g., *beautiful*1*a* is similar to *pretty*1*a*).
10. *Attribute*: a noun *X* is an attribute for which an adjective *Y* expresses a value (e.g., *hot*1*a* is a value of *temperature*1 *n*).
11. *See also*: this is a relation of relatedness between adjectives (e.g., *autiful*1*a* is related to *attractive*1*a* through the see also relation).[8]

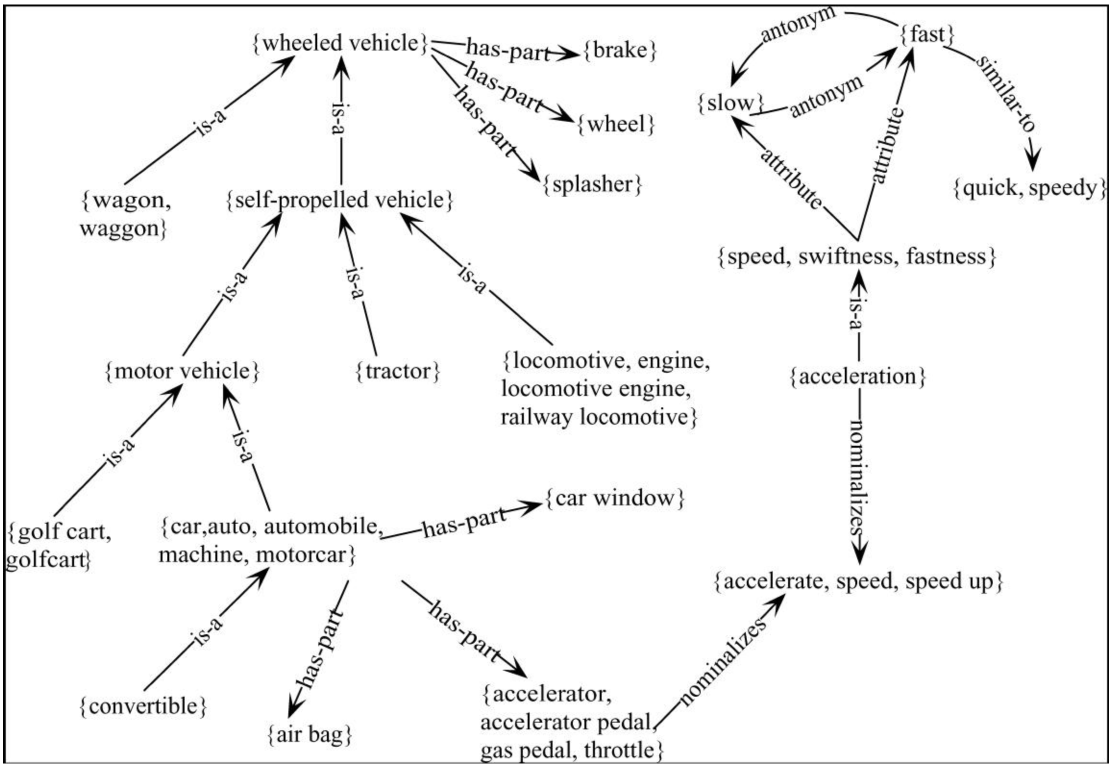

Fig. 2. An excerpt of the WordNet semantic network.

2.2 Algorithms

According to whether additional training corpora are used, WSD algorithms can be roughly classified into supervised and unsupervised categories.

1. Supervised WSD: these approaches use machine-learning techniques to learn a classifier from labeled training sets, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class);
2. Unsupervised WSD : these approaches does not require a training corpus and needs less computing time and power. It is suitable for online machine translation and information retrieval. However, it theoretically has worse performance than the supervised approach because it relies on less knowledge.[1]

3. Basis of WSD task

In computational linguistics, **word sense disambiguation** (WSD) is the process of identifying which sense of a word is used in any given sentence, when the word has a number of distinct senses. For example, consider two examples of the distinct senses that exist for the (written) word *bass*:

1. a type of fish
2. tones of low frequency

and the sentences:

1. *I went fishing for some sea bass.*
2. *The bass line of the song is too weak.*

To a human, it is obvious that the first sentence is using the word *bass*, as in the former sense above and in the second sentence, the word *bass* is being used as in the latter sense below.
If we disregard the punctuation, we can view a text *T* as a sequence of words ($w1$, $w2$, . . . , wn), and we can formally describe WSD as the task of assigning the appropriate sense(s) to all or some of the words in *T*, that is, to identify a mapping *A* from words to senses, such that $A(i) \subseteq SensesD(wi)$, where $SensesD(wi)$ is the set of senses encoded in a dictionary *D* for word wi , and $A(i)$ is that subset of the senses of wi which are appropriate in the context *T*. The mapping *A* can assign more than one sense to each word $wi \in T$, although typically only the most appropriate sense is selected, that is, $| A(i) |= 1$.
WSD can be viewed as a classification task: word senses are the *classes*, and an *automatic classification method* is used to assign each occurrence of a word to one or more classes based on the evidence from the *context* and from *external knowledge sources*.[8]

4. Towards a hybrid semantic approach to WSD

Word sense disambiguation is the ability to computationally determine which sense of a word is activated by its use in a particular context. It is usually performed on one or more texts besides some learning techniques, according to the adopted algorithm. Each way has its own contribution to the process of WSD we believe if we combine the learning algorithms with semantic features and some previously defined hypothesis we will have better performance and more reliable results.

4.1 Semantic similarity SS

Semantic similarity is an important topic in natural language processing (NLP). It has also been subject to studies in Cognitive Science and Artificial Intelligence. It is a kind of semantic relatedness defining a resemblance.
First studies in this area dateback to Quilian's semantic memory model (Quilian,1968) where the number of hops between nodes of concepts in the hierarchical network specifies the similarity or difference of concepts. Wu and Palmer's semantic similarity measure (WUP) was based on the path length between concepts located in a taxonomy (Wu and Palmer, 1994)[9].
Nowadays, computational models of similarity are being included in many software applications with the intent of making these applications seem more intelligent or even creative. Application areas of semantic similarity include word sense disambiguation (WSD), information retrieval, etc.
Information Content (IC) is an important dimension of word knowledge when assessing the similarity of two terms or word senses. The conventional way of measuring the IC of word senses is to combine knowledge of their hierarchical structure from an ontology like WordNet, which is generally a hand-crafted lexical database [4] as mentioned above.
The information content of a concept *c* can be quantified as negative the log likelihood, — log p(c). This assumption is made by Resnik [2]for the first time were he suggested that the quantifying information content in this way means : as probability increases, informativeness decreases, so the more abstract a concept, the lower its information content. Moreover, if there is a unique top concept, its information content is 0. In this way he suggested that "the quantitative characterization of information provides a new way to measure semantic similarity.

The more information two concepts share in common, the more similar they are, and the information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy".[2]
According to Resnik in [2], SS depends on the amount of information two concepts have in common, this shared information is given by the Most Specific Common Abstraction (MSCA) that subsumes both concepts. In order to find a quantitive value of shared information we must first discover the MSCA, if one does not exist then the two concepts are maximally dissimilar, otherwise the shared information is equal to the IC value of the MSCA. Formally, semantic similarity is defined as:

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} ic_{res}(c)$$

Seco in [3] creates anew improved equation for calculating IC without using statistical resources other than WorldNet. Their work based on the assumption that, "WordNet can be used as a statistical resource with no need for external ones". They argue that the WordNet taxonomy may be innovatively exploited to produce the IC values needed for SS calculations. They obtained IC values according to the taxonomic structure of WordNet, as that taxonomy organized in a meaningful and principled way, where concepts with many hyponyms convey less information than concepts that are leaves. In fact they argue that the more hyponyms a concept has the less information it expresses. So concepts that are leaf nodes are the most specified in the taxonomy and the information they express is maximal.
They presented in [3] a novel equation of IC that is completely derived from WordNet without the need for external resources from which statistical data is gathered. The test of their method showed that this new equation delivers better results when they substitute new IC values (with the corpus derived ones) in previously established formulations of SS. The formula he derived like the following

$$ic_{wn}(c) = \frac{log(\frac{hypo(c)+1}{max_{wn}})}{log(\frac{1}{max_{wn}})} = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$$

where the function *hypo* returns the number of hyponyms of a given concept and *maxwn* is a constant that is set to the maximum number of concepts that exist in the taxonomy.
We suggest the adoption of this measure of semantic similarity in the structural semantic interconnection algorithm SSI which is funded by the Italian Legenda MIUR project and adopted by the OntoLearn project which is partly funded by the INTEROP IST-508011 Network of Excellence http://www.lap.u-bordeaux1.fr/interopnoe/

4.2 Structural Semantic Interconnection algorithm SSI

OntoLearn is an ontology learning project, which is based on machine learning and text mining approach which learns domain concepts and taxonomic relations from input data [1]. OntoLearn combines natural language processing and statistical techniques for terminology extraction. Their Structural Semantic Interconnections (SSI) approach employs a syntactic pattern-matching algorithm to perform word sense disambiguation. A compositional interpretation technique is applied whereby the meaning of complex terms can be derived from its components Semantic relations between the components of a complex concept are determined using Context grammar rules. After a complex term is semantically interpreted, it is then integrated into the initial ontology (WordNet) and linked to a suitable parent node.

Structural semantic interconnections (SSI), is a WSD algorithm that uses graphs to describe the objects to analyze (word senses) and a context free grammar to detect relevant semantic patterns between graphs. Sense classification is based on the number and type of detected interconnections. The graph representation of word senses is automatically built from several available resources, such as lexicalized ontology, collocation inventories, annotated corpora, and glossaries.

To perform semantic disambiguation, they use a number of lexical knowledge bases (LKB) , like WordNet and annotated corpora(Texts provide examples of word sense usages in context.), and the word sense disambiguation (WSD) algorithm structural semantic interconnection SSI.

A graph-based representation of word senses provides better description through a set of structured features. For example, the representations of the WordNet definition of senses #1 (vehicle) and #2 (connector) of bus, where nodes represent concepts (WordNet synsets) and edges are semantic relations.

Representation shows the semantic interrelationships for the words in the definition. And depending on that representation for alternative word senses in a context, *disambiguation can be seen as the task of detecting certain "meaningful" interconnecting patterns among such graphs*.

The graphs automatically built using a variety of knowledge source and called *semantic graphs* as shown in figure 3.

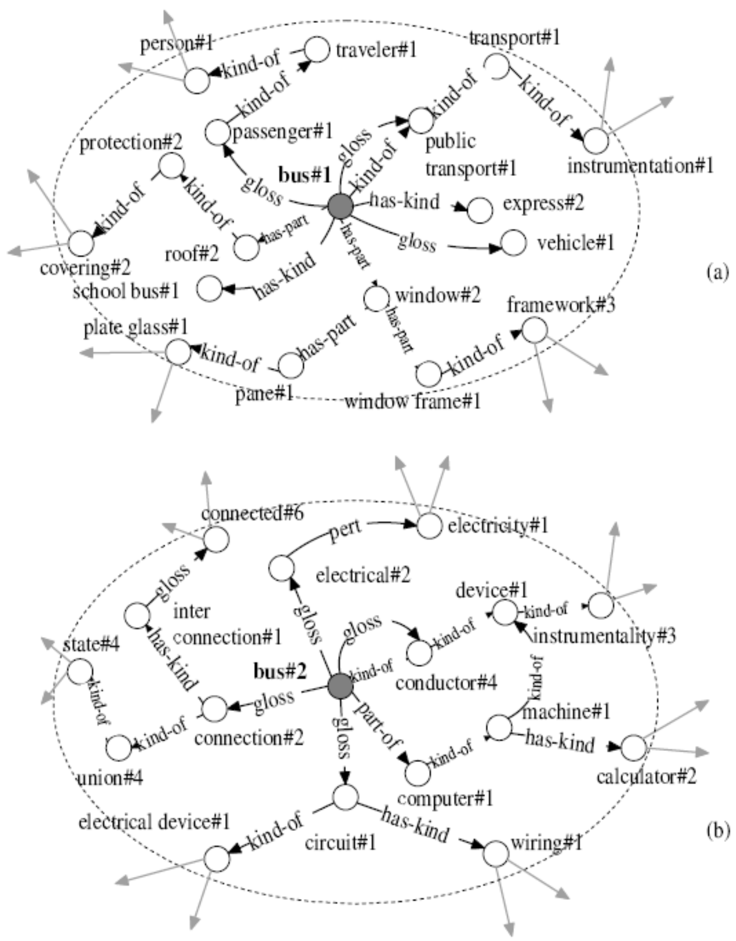

Figure 3 semantic graph of the word "bus" in two senses

This assumption of course needs a number of linguistics tools such as a context-free grammar, to accomplish the disambiguation process. The used CFG is build for a specific domain and to specify the type of patterns that are the best indicators of a semantic interrelationship and to select the appropriate sense configurations accordingly (figure 4).

$S_G \rightarrow S_s \mid S_g$	*(all the heuristics)*
$S_s \rightarrow S_1 \mid S_2 \mid S_3$	*(simple heuristics)*
$S_1 \rightarrow E_1 S_1 \mid E_1$	*(hyperonymy/meronymy)*
$E_1 \rightarrow e_{\text{kind-of}} \mid e_{\text{part-of}}$	
$S_2 \rightarrow E_2 S_2 \mid E_2$	*(hyponymy/holonymy)*
$E_2 \rightarrow e_{\text{has-kind}} \mid e_{\text{has-part}}$	
$S_3 \rightarrow e_{\text{kind-of}} S_3 e_{\text{has-kind}} \mid e_{\text{kind-of}} e_{\text{has-kind}}$	*(parallelism)*
$S_g \rightarrow e_{\text{gloss}} S_s \mid S_4 \mid S_5$	*(gloss)*
$S_4 \rightarrow e_{\text{gloss}} \mid e_{\text{topic}}$	*(gloss, context)*
$S_5 \rightarrow e_{\text{gloss}} e_{\text{is-in-gloss}}$	*(gloss+gloss^{-1})*

Figure 4 part of CFG used by SSI algorithm

The classification problem can be stated in SSI as follows:

- T (the lexical context) is a list of related terms.
- t is a term in T to be disambiguated.
- St1; St2; . . . ; Stn , are structural specifications of the possible concepts for t (precisely, semantic graphs).
- I (the semantic context) is a list of structural specifications of the concepts associated to (some of) the terms in T. I includes either one or no specifications for each term in T.
- G is a grammar describing relevant relations between structural specifications (precisely, semantic interconnections among graphs).
- Determine how well the structural specifications in I match that of each of St1; St2; . . . ; Stn , using G.
- Select the best matching St

Structural representations are graphs, as previously detailed. The SSI algorithm consists of an initialization step and an iterative step. The input is a list of co-occurring terms *T* = [*t*1, . . . , *tn*] and a list of associated senses *I* = [*St*1 , . . . , *Stn*], that is, the semantic interpretation of *T*, where *Sti* is either the chosen sense for or the empty set (i.e., the term is not yet disambiguated). A set of *pending* terms is also maintained, *P={ti|Sti* =0 *}*. *I* is referred to as the semantic context of *T* and is used, at each step, to disambiguate new terms in *P*.
The process of compositional interpretation associates the appropriate WordNet synset Sk with each word tk in T. The sense of T is hence compositionally defined as

S(T) = [Sk|Sk € Synsets(tk),tk € T]

where *Synsets*(t*k*) is the set of senses provided by WordNet for word t*k*, for instance:

S("transport company__) = [*{transportation*#4, *shipping*#1, *transport*#3*}*, *{company*#1*}*]

The algorithm works in an iterative way, so that at each stage either at least one term is removed from *P* (i.e., at least one pending term is disambiguated) or the procedure stops because no more terms can be disambiguated. The output is the updated list *I* of senses associated with the input terms *T*.

Initially, the list *I* includes the senses of monosemous terms in *T*. During a generic iteration, the algorithm selects those terms *t* in *P* showing an interconnection between at least one sense *S* of *t* and one or more senses in *I*. The likelihood that a sense *S* will be the correct interpretation of *t*, given the semantic context *I*, is estimated by the function *fI* : *Synsets* × *T* →_, where Synsets is the set of all the concepts in WordNet, and defined as follows:

$$f_I(S,t) = \begin{cases} \rho(\{\varphi(S,S') \mid S' \in I\}) & \textit{if } S \in Senses(t), \\ 0 & \text{otherwise,} \end{cases}$$

Where *Senses*(*t*) is the subset of synsets in WordNet associated with the term *t*, and

$$\phi(S,S') = \rho'(\{w(e_1,e_2,\ldots,e_n) \mid S \xrightarrow{e_1} S_1 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} S_{n-1} \xrightarrow{e_n} S'\})$$

that is, a function (ρ') of the weights (*w*) of each path connecting *S* with *S'*, where *S* and *S'* are represented by semantic graphs. A semantic path between two senses *S* and *S'*

$$S \xrightarrow{e_1} S_1 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} S_{n-1} \xrightarrow{e_n} S',$$

is represented by a sequence of edge labels *e*1, *e*2, . . . , *en*.

4.3 The hybrid approach

As we can see the main criteria to judge the right diagnosis of the right sense depends (among others) on the possible direct semantic interconnections among the semantic nets. Although this is a good factor since we are depending on a the context information provided by the co occurrence words, but the main thing here is that we are not using the word order provided in the syntactic information, i.e. we are not using the roles dedicated to the different words in the sentence. Also we are not using the previously provided information in the ontology used like WordNet about the semantic similarity SS that can be found depending on the information content IC.

The SS measure can work to a good factor especially with semantic nets where the structural representation for each word will detail the semantic features. Also the SS will putdown the need of CFG which is (in SSI ago.) the main limitation for this approach. What we suggest here, is instead of trying to find the possible connection between the semantic nets (taken maximally in branching up to three nodes) which in many times might not be found and consuming in terms of processing , why not trying to find the possible matching semantic nets for the senses of the words in context depending on other similar samples in the domain which of course will talk about the same discourse.

If we came to believe the hypothesis of the one-sense-per discourse , which proven to be correct 98% as stated in [5], then this hypothesis can be used to add source of constraint for improving the performance of the word-sense disambiguation algorithm. So depending on this hypothesis we can be sure that the SS method will be straight to the goal in finding the right sense after the first iteration of disambiguation.

The implementation of the hybrid system that we propose will follow the following steps

1. T (the lexical context) is a list of related terms.
2. t is a term in T to be disambiguated.

3. St1; St2; . . . ; Stn , are structural specifications of the possible concepts for t (precisely, semantic graphs).
4. I (the semantic context) is a list of structural specifications of the concepts associated to (some of) the terms in T. I includes either one or no specifications for each term in T .
5. G is a grammar describing relevant relations between structural specifications (precisely, semantic interconnections among graphs).
6. Determine how well the structural specifications in I match that of each of St1; St2; . . . ; Stn , using G.
7. Calculate the IC of each sense in St1; St2; . . . ; Stn. Consider this set1
8. Calculate the IC of each sense in I. Consider this set2
9. In both cases set1 and set2 the following equation are used

$$ic_{wn}(c) = \frac{log(\frac{hypo(c)+1}{max_{wn}})}{log(\frac{1}{max_{wn}})} = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$$

10. Calculate the best semantic similarity betweenset1and set2 according to equation

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} ic_{res}(c)$$

11. Compare the results of step6 and step 10. The greater the semantic similarity measure exist must be supported by the best matching factor of step 6.
12. Select the best matching St

In fact after some rounds of testing the previous algorithm we improved it later to get maximal advantages, as follows
The implementation of the hybrid system that we propose will follow the following steps

1. T (the lexical context) is a list of related terms.
2. t is a term in T to be disambiguated.
3. St1; St2; . . . ; Stn , are the possible concepts for t .
4. I (the semantic context) is a list of the concepts associated to (some of) the terms in T. I includes either one or no specifications for each term in T
5. Calculate the IC of each sense in St1; St2; . . . ; Stn. Consider this set1
6. Calculate the IC of each sense in I. Consider this set2
   In both cases set1 and set2 the following equation are used

$$ic_{wn}(c) = \frac{log(\frac{hypo(c)+1}{max_{wn}})}{log(\frac{1}{max_{wn}})} = 1 - \frac{log(hypo(c)+1)}{log(max_{wn})}$$

7. Calculate the best semantic similarity between set1and set2 according to equation $sim_{res}(c1,c2)$, in which a set of similarities should be retrieved, call this set 3

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} ic_{res}(c)$$

8. If the best SS got then go to step12
9. build the structural specifications of the concepts in set3 (semantic graphs).
10. G is a grammar describing relevant relations between structural specifications (precisely, semantic interconnections among graphs).

11. Determine how well the structural specifications in set3 connected to each other , using G.
12. Select the best matching St

The SS here acts as an additional source of assurance for improving the performance of the word-sense disambiguation algorithm. In this case rather than tagging each instance of a polysemous word one-by-one in a corpus, we can select discourses with large numbers of the polysemous word of interest and tag all of the instances in one act. Admittedly, this procedure will introduce a small error rate since the one-sense-per-discourse tendency is not quite perfect, of assigning the right sense depending on only WordNet and discarding all other statistical resources.
Another result from the previous algorithm is that we can add more value to the judgment of the results in the original SSI algorithm. Or even using it for the cases were thre is no disambiguation at all

We are achieving lies in the following boundaries

1- Reducing the amount of knowledge required in the sense of using semantic similarity to guide the tagging process based on one sense per discourse. Also making use of IC of Seco which eliminates the use of statistical corpus.
2- Learning knowledge by making use of the already existing knowledge sources that provide semantic information regarding concepts and their associated words (semantic similarity).
3- In case of our second application of the SSI algorithm we depend mainly on SS giving less role to semantic nets and performing less processing making use of the advantage that in the one discourse most of the words will have clear (may be unique sense).So there will be obvious cases in which no semantic nets will be required.
4- Using the existing knowledge sources to build learning examples automatically in restricted manner.
5- The learning algorithm should make use of both labeled knowledge sources and the generated semantic information in the operation of assigning concepts, depending mainly on the semantic information provided in WordNet.
6- The process does not depend on guessing rather than searching and matching the resultant senses according to provided information**.**

6. RECOMMENDATIONS

Word Sense Disambiguation WSD is an open problem in Natural Language Processing. Its solution impacts many tasks such as MT, IR ,OL, and others. Since then there were many techniques and method that implement different AI approaches to achieve the best result. The hybrid approaches proved to be more effective were they make use of the advantages of more than one algorithm or methods to enhance the output.
The semantic features analysis have been in the field of NLP for long time and its proven to be more effective when used, as it represent the intended usage of the item. We tried to use the semantic similarity SS and its power in the one sense per discourse hypothesis to enhance the performance of the SSI in OntoLraern project.
We make use of SS which in turn depends on what already had been build (WordNet knowledge), and combine it with one sense per discourse hypothesis. The power we

got here is that since the context is dedicated in most cases to a unique discourse then the power of similarity will increase and in turn we will be able to get more similar examples of word senses that makes it easier and more effective especially when we can represent the semantic features of the words as represented in WordNet.

Also our approach makes use of less CFG, since most of the hard work for assigning the right concept (sense) is done by SS. This complication (i.e. forming the right CFG) considered the main limitation in SSI algorithm. They stated that "The classification task in a structural pattern recognition system is implemented through the use of grammars that embody precise criteria to discriminate among different classes. The drawback of this approach is that grammars are by their very nature application and domain specific"[1].

Regarding the knowledge sources adopted by WSD systems, in recent years, the results of many research efforts for the construction of online lexical knowledge repositories, ontologies and glossaries became available creating new opportunities for knowledge-based sense disambiguation methods. The problem is that these resources are often heterogeneous, midway formal, and sometimes inconsistent. Despite these problems, we believe that these resources still represent an important and necessary tool for WSD and can be enhanced through better formalization. Also we believe that making use of SS in automatic generating of such resources would yield better resources with less efforts and huge amount of information,in case of good seeding of the original semantic constrains.

In general we can identify three trends with respect to the future improvement of WSD algorithms. They are, the use of word order information assigning the right sense, addressing the relative importance of semantic features in the model by some elegant techniques, and the increase of the size of training data.

7. References

1. R. Navigli, and P. Velardi, “Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites”, *Computational Linguistic*, *MIT Press*, 2004, (50)2.
2. Philip Resnik, ‘Using information content to evaluate semantic similarity in a taxonomy’, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* pp. 448–453, (1995).
3. N. Seco, T. Veale, and J. Hayes, “An Intrinsic Information Content Metric for Semantic Similarity in WordNet”, In *Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence*, Valencia, Spain, 2004.
4. C. Fellbaum , "WordNet: An Electronic Lexical Database", MIT Press, 1998.
5. *William A. Gale,Kenneth W. Church,David Yarowsky* ,One Sense Per Discourse , 1992
6. Roberto Navigli and Paola Velardi , “Structural Semantic Interconnections:A Knowledge-Based Approach to Word Sense Disambiguation”, IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 7, july 2005
7. Nanc Ide and Jean Véronis, “ Word Sense Disambiguation: The State of the Art”, Computational Linguistics, 1998, **24**(1).
8. Roberto Navigli, “Word Sense Disambiguation: A Survey”,ACM Computing Surveys, Vol. 41, No. 2, Article 10, 2009.

9. Ergin Altintas , Elif Karsligil , Vedat Coskun , "A new semantic similarity measure evaluated in word sense disambiguation",2006.
10. Yoong Keok Lee and Hwee Tou Ng , "An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation", Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 41-48. Association for Computational Linguistics.
11. Rada Mihalcea and Dan I_ Moldovan , "Word Sense Disambiguation based on Semantic Density",1996.

كيفية كشف غموض المغزى المراد للكلمات بالاعتماد على خواص المعنى للكلمة

آمال صالح حسن

جامعة السلطان قابوس ، كلية العلوم ، قسم الحاسب الآلي ، عمان

amaalh@squ.edu.om

واحدة من أكثر المشاكل التي تواجه الباحث في مجال معالجة اللغات الطبيعية هي كشف المغزى المراد بكلمة معينة في نص ذا موضوع محدد. هذا الموضوع اخذ حيز كبير في مجال الحسابات اللغوية التي تنفذ بواسطة الكومبيوتر من اجل معالجة اللغات الطبيعية. بالحقيقة أن معرفة مغزى الكلمات بحاجة إلى قاعدة معلومات كبيرة و التي يجب أن تكون منظمة و مرتبة بطريقة تمكن الرجوع إليها و استخلاص المعلومات منها بسهولة.

هذه المشكلة(معرفة مغزى الكلمات الصحيح في النص) مثلت عقدة لاستعمال اللغات الطبيعية ضمن مدى تطبيقات واسعة.و هذا أدى بالباحثين إلى محاولات كثيرة و متعددة لإيجاد طرق جديدة لإيجاد مغزى الكلمة و باستعمال تقنيات مختلفة ،منها الطرق الإحصائية و الخوارزميات الذكية القابلة للتعلم و مصادر المعلومات المتطورة ذات الطبيعة المحددة و التي نقصد بها ان مغزى الكلمات فيها واضح و لكن يستفاد منها لتوضيح مصادر أخرى.

إن أفضل الطرق هي التي تجمع بين بعض ما تم ذكره بتوليفة مناسبة تؤدي إلى استخلاص مغزى الكلمة بسهولة و بدقة أكثر و هذا هو الذي سنتكلم عنه في بحثنا و سنستعرض مثال لطريقة معينة و كيفية أن استخدام خواص المعنى ممكن أن تؤدي إلى نتائج أفضل عند التطبيق.