2010 IEEE Fourth International Conference on Semantic Computing

A Word Sense Disambiguation Approach for Converting Natural Language Text into a Common Semantic Description

Francisco Tacoa
Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan
tacoa@mi.ci.i.u-tokyo.ac.jp

Hiroshi Uchida
UNDL Foundation
Tokyo, Japan
uchida@undl.org

Mitsuru Ishizuka
Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan
ishizuka@i.u-tokyo.ac.jp

***Abstract*—Concept Description Language (CDL) is a common language that represents the semantics of content in a simple and structured manner. In particular, it is intended to describe Natural Language (NL) texts in a format that can be understood and processed by computers. Since words with multiple meanings can be found from texts, it becomes necessary to perform Word Sense Disambiguation (WSD) in order to achieve a correct representation. This paper presents a WSD approach that determines best candidates for word meanings and contributes to a semi-automatic conversion of NL into CDL. We perform preliminary experiments by evaluating the approach with some test sentences and comparing with other WSD methods. Results suggest that the existence of a proper correspondence of syntactic and semantic relations for the WSD process may lead to an accurate conversion to CDL.**

***Keywords*-Semantic computing; Concept description; Natural language text;**

I. Introduction

A. Concept Description Language (CDL)

CDL is a computer language proposed by the Institute of Semantic Computing (ISeC)[1], to perform Semantic Computing (SeC). It describes a wide variety of representation media and content, as well as conceptualization of their meaning, in a common format [13]. Some of its purposes are: to represent semantic meaning of texts, to overcome language barriers, and to realize machine understandability.

1) CDL Structure: There is the notion of specifying a different CDL for description of every aspect in the real world.

CDL.nl is a set of different CDLs for description of concepts from natural languages. In consequence, there is "CDL.eng" for concepts given in English, "CDL.jpn" for concepts in Japanese, "CDL.chi" for concepts in Chinese, and so on.

"CDL.unl" is the CDL version of the Universal Networking Language (UNL) [11]. Details of UNL can be seen in Section II.

[1]http://www.instsec.org/ – (in Japanese)

More examples of CDL include: "CDL.math" for description of mathematical formula, "CDL.prog" for programming languages, "CDL.movie" and CDL.music for description of different media types, etc.

Since this work covers aspects related to natural language, only CDL.nl will be employed.

2) CDL Representation: CDL includes the following basic elements, which can be represented in either text or graph notations:

- "Entity", to indicate concepts.
- "Relation", to indicate a link between two concepts.
- "Attribute", to describe logical characters and properties of concepts.

B. Word Sense Disambiguation (WSD)

WSD is the process of selecting the correct sense for a word with multiple meanings. It is an intermediate and important step that allows other tasks, such as Machine Translation, Information Retrieval and Text Mining, to generate correct results, especially when natural language sentences could be interpreted in several ways. Details about categories and approaches in this area can be seen in [1].

In order to construct an appropriate semantic description for a sentence, it is necessary that all ambiguities are resolved. Therefore, WSD becomes an important task prior to the conversion to CDL format.

The approach proposed in this paper contributes to the disambiguation of word meanings from natural language English texts in a semi-automatic way. This means that the algorithm will calculate the best candidates for word meanings and will indicate them to the user, but the user will have the final decision for selecting those that he/she considers correct.

In this paper, Section II presents some background and previous work, followed by the details about our approach in Section III and a comparative method in Section IV; some aspects about semi-automatic conversion from NL to CDL are discussed in Section V; the CDL Graphical Editor is presented in Section VI; Section VII contains preliminary

978-0-7695-4154-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICSC.2010.66

experiments and results; and finally, conclusions and future work are presented in Section VIII.

II. Background and Related Work

A. Universal Networking Language (UNL)

UNL is a language that describes semantics of electronic contents, originally developed in 1996 by the Institute of Advanced Studies of the United Nations University (UNU/IAS)[2], but in 2001, research and development activities were transferred to the Universal Networking Digital Language (UNDL) Foundation[3].

According to [11], UNL is an artificial language that allows computers to process information and knowledge, regardless of language limitations. Natural language sentences are represented as a semantic network, where nodes represent concepts and arcs represent relations between concepts. These concepts, which are commonly referred to as Universal Words (or UWs for short), can also be annotated by attributes to provide additional information based on the circumstances under which they are being used.

UWs are divided into four types:

- Basic UWs, which are headwords that do not indicate any constraints, used for the representation of unambiguous words. For example:
  - accelerometer
  - information
  - quantity
- Restricted UWs, which are headwords with a specific constraint or constraints list indicated inside parentheses. For example, the word "state" has the following constraints:
  - state(icl>express(agt>thing,gol>person,obj>thing))
  - state(icl>country)
  - state(icl>region)
  - state(icl>abstract thing)
  - state(icl>government)
- Extra UWs, another type of Restricted UW but applied for foreign-language words, for example:
  - ikebana(icl>flower arrangement)
  - samba(icl>dance)
  - souffle(icl>food)
- Temporary UWs, which are not necessary to define, such as: "1234", "xyz".
  Phone numbers and e-mail addresses are examples of this type of UW.

Currently, UWs have been created for up to 16 languages: Arabic, Armenian, Bengali, Chinese (simplified), English, French, German, Indonesian, Italian, Japanese, Latvian, Mongolian, Portuguese, Russian, Spanish, and Thai.

[2]http://www.ias.unu.edu/
[3]http://www.undl.org/

UNL also includes a component called Universal Networking Language Knowledge Base (UNLKB)[4], which provides a lexicon based on UWs. This lexicon is an ontology that groups six type of UWs, as shown in Figure 1.

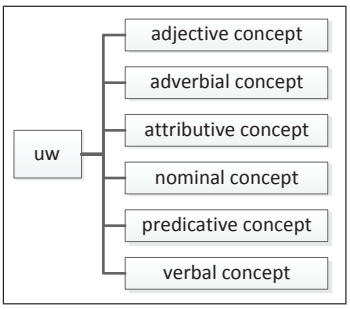

Figure 1. Different types of UW in UNL Ontology.

B. Common Web Language (CWL) Platform

CWL Platform[5,6] is a web-based application that performs machine translation for English, Japanese, French, Russian, Spanish and Arabic languages.

CWL is designed to be used as descriptor for meta-data and contents of web pages for breaking language barriers and enable computers to process web information semantically. It can be represented in any of the following formats:

- CWL.unl: language based on UNL, intended for multilingualism.
- CWL.cdl: language based on CDL, intended for compatibility with semantic computing systems.
- CWL.rdf: language based on the Resource Description Framework (RDF/OWL)[7] format, intended for working with various data navigation and aggregation systems.

The CWL Platform consists of the following modules:

- CWL Editor: for inputting natural language texts, and for selection of the most appropriate word meaning for words.
- CWL Converter: for conversion of NL into CDL, UNL, or RDF.
- UNL Enconverter: for NL $\rightarrow$ UNL conversion.
- UNL Deconverter: for UNL $\rightarrow$ NL conversion.

UNL system is used as the support that provides the CWL Platform with vocabulary and semantic relations.

The CWL Editor module allows users to select manually word senses, but this process will constitute a heavy load of work in the case of many sentences.

[4]http://www.undl.org/unlsys/uw/UNLKB.htm
[5]http://www.undl.org:8080/cwl/
[6]http://www.w3.org/2005/Incubator/cwl/XGR-cwl/
[7]http://www.w3.org/RDF/

C. Word Sense Disambiguation

1) Graph-Based Method: The work presented in [10] proposes an unsupervised graph-based method for disambiguation of words. A graph where nodes represent word meanings and edges represent relations between two nodes is constructed. Next, after applying a semantic similarity measure, nodes' weight are calculated by a graph-based centrality algorithm. As result, nodes with highest weight will be considered the correct meanings for their respective words.

2) Semantic Role Labeling for WSD: Previous works in [5] and [6] combined WSD and Semantic Role Labeling (SRL), to improve the precision of Question Answering (QA) and Information Retrieval (IR) systems. First, they perform disambiguation of verb senses; next, disambiguation of arguments; and finally, disambiguation of semantic roles.

III. WSD by Analysis of Semantic Relations

The purpose of the WSD approach proposed here is to calculate the best candidate for word meanings, based on the semantic relations and noun classes given by verb candidates. This method runs as Word-to-Class selectional preference [1], since UWs for verbs in UNL contain arguments as follows:

$$\texttt{verb}\big(\texttt{rel}_1 > \texttt{noun\_class}_1, \ldots, \texttt{rel}_n > \texttt{noun\_class}_n\big)$$

A. Tools

The following tools are used for the approach:

1) Data source: Provided by UNLKB (see Section II-A). Restricted UWs are the most used of all UWs as they contain logical restrictions (in this case, a noun class) labeled with semantic relations (i.e., "agt" for agent, "obj" for object, "src" for initial state, "gol" for final state, and so on) necessary to calculate the best candidates. A full list of semantic relations used in CDL is available in [11].

2) Syntactic parser: Provided by the Stanford Parser[8]. It is used for syntactic analysis of sentences.

Consider the following sample sentences:

```
(A)  "John loves Alice."
(B)  "John loves cars."
```

For sentence (A), syntactic relations should be as follows:

$$\texttt{subj}(\texttt{love}, \texttt{John})$$
$$\texttt{obj}(\texttt{love}, \texttt{Alice})$$

For sentence (B), syntactic relations are:

$$\texttt{subj}(\texttt{love}, \texttt{John})$$
$$\texttt{obj}(\texttt{love}, \texttt{car})$$

Besides syntactic dependencies, there are more elements used in the method, such as lemma and part-of-speech tags.

A simple sentence that contains ambiguous noun and verb, denoted as `(C)`, is available in Section III-C.

[8]http://www-nlp.stanford.edu/software/lex-parser.shtml

B. Overview

Best candidates for word meaning are determined by:

- the number of semantic relations for which a syntactic relation exists, and
- the distance between nouns and noun classes. Distance is calculated by edge counting.

Syntactic analysis helps to distinguish which words are connected with which relations.

For instance, we will analyze the previous sample sentences. UNL Ontology contains unambiguous UWs for words "John", "Alice" and "car" as indicated in Figure 2:

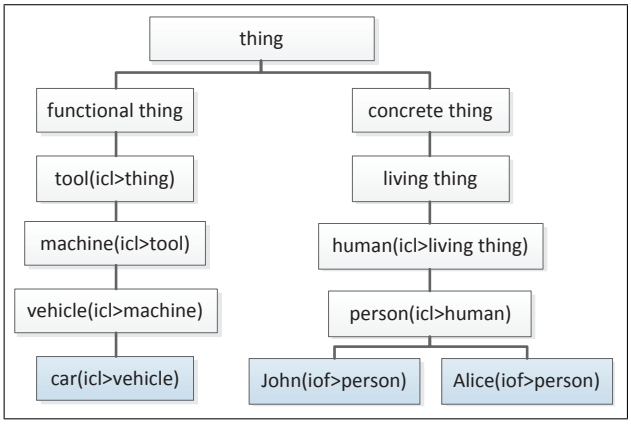

Figure 2. UWs in UNL Ontology.

As for verb "love", UNL Ontology defines the following UWs, which constitute the possible best candidates in our approach:

(1) $\texttt{love}(\texttt{agt} > \texttt{person})$

(2) $\texttt{love}(\texttt{agt} > \texttt{person}, \texttt{obj} > \texttt{person})$

(3) $\texttt{love}(\texttt{agt} > \texttt{person}, \texttt{obj} > \texttt{thing})$

Verb candidate (1) requires an agent-type UW under the noun class "person"; verb candidate (2) requires both agent and object-type UWs under the noun class "person"; and verb candidate (3) requires an agent-type UW under the noun class "person" and an object-type UW under the noun class "thing".

For syntactic relations "John" is the subject, and becomes agent for semantic relation. Similarly, "Alice" and "cars" are the objects of the verb in their respective sentences, for both syntactic and semantic relations.

Tables I and II show the criteria for best candidates calculation. First condition is to have the maximum Total Valid Relations (TVR); and second condition is that candidates who satisfied the first condition, must have the minimum value of the sum of all distances between nouns and noun classes given by them. As result, by applying these conditions the best candidates for verb "love" are (2) and (3) for sentences (A) and (B), respectively.

Table I
BEST CANDIDATE CALCULATION FOR VERB IN SENTENCE (A) (DETERMINED TO BE `love(agt>person,obj>person)`).

Verb candidate (No.)	Is valid agt?	Is valid obj?	TVR	$\sum Dist$
(1)	Yes	—	1	—
(2)	**Yes**	**Yes**	**2**	**2**
(3)	Yes	Yes	2	6

Table II
BEST CANDIDATE CALCULATION FOR VERB IN SENTENCE (B) (DETERMINED TO BE `love(agt>person,obj>thing)`).

Verb candidate (No.)	Is valid agt?	Is valid obj?	TVR	$\sum Dist$
(1)	Yes	—	1	—
(2)	Yes	No	1	—
(3)	**Yes**	**Yes**	**2**	**7**

For nouns, the best candidates will be those that contributed to the calculation of verb best candidate. In other words, verbs best candidates will determine automatically nouns best candidates.

C. Method

As explained in the Section III-B, the distance between nouns and noun classes is determined by the total of edges that separate them. This is applied in equations 1 and 4 from the following explanation. In detail, the WSD method goes through four main steps:

1) **Syntactic analysis:** Get syntactic information of words, such as lemma, part of speech, and dependency relations. The syntactic dependency relations that are relevant in this case are of verb-noun and noun-noun types, since the method only calculates best candidates for nouns and verbs.
2) **Extraction of verb and noun candidates:** Words lemmas are used to extract UWs from UNL Ontology. For each lemma, multiple results can be returned, depending on the number and type of semantic relations that the UWs contain. This difference of relations is what makes possible to have multiple candidates.
3) **Analysis of verb-noun relations:** This step is divided into three sub-steps:
   a) *Filter verb candidates:* Consider only those candidates whose semantic relations have the corresponding syntactic relations.
   b) *Determine best candidates for nouns:* Best candidates for nouns can be calculated by their distance to the corresponding noun class, through the equation 1:

$$BC_{Noun} = min\big(dist\,(NC, Nc_1), \ldots, \quad (1)$$
$$dist\,(NC, Nc_n)\big)$$

   where NC is the noun class and Nc_i is the noun candidate. This equation is repeated for each noun candidate.
   c) *Determine best candidate for verbs:* Best candidates for verbs are calculated by equations 2 and 3. However, 3 will be applied only if 2 returns more than one possible candidate:

$$BC_{Verb} = max\,(TVR_1, TVR_2, ..., TVR_n) \quad (2)$$

$$BC_{Verb} = min\Big(\sum Dist_{maxTVR_1}, \ldots, \quad (3)$$
$$\sum Dist_{maxTVR_n}\Big)$$

   where TVR_i means the Total Valid Relations for candidate i, and $Dist_{maxTVR_i}$ represents the sum of the distances for the candidate i with maximum TVR.
4) **Analysis of noun-noun relations:** Since not all nouns are connected to verbs in the syntactic dependencies, this step aims to calculate best candidates for nouns that still have not been processed. The syntactic relations determine which nouns are connected to each other.
   Instead of noun class (equation 1), best candidates of already processed nouns are used to calculate distance (equation 4):

$$BC_{Noun_2} = min\big(dist\,(BC_{N_1}, N_2c_1), ..., \quad (4)$$
$$dist\,(BC_{N_1}, N_2c_n)\big)$$

   where BC_{N_1} is the best candidate of the already processed noun, and N_2c_i represents each candidate of the noun for which the best candidate is being calculated.

Consider the following example:

`(C) "John has the list of instructions."`

The following are the relevant syntactic relations provided by Stanford Parser for this sentence:

`nsubj(have, John)`
`dobj(have, list)`
`prep_of(list, instruction)`

First, best candidates are calculated for words in the verb-noun relations (applying the Step 3). In this case, words from “nsubj” and “dobj” syntactic relations will be

Table III
BEST CANDIDATE FOR VERB IN SENTENCE (C)
(have(agt>person,obj>thing)).

Verb candidate (No.)	TVR	$\sum Dist$
(4)	**2**	**5**
(5)	2	9

Table IV
BEST CANDIDATE FOR THE NOUN "INSTRUCTION" IN SENTENCE (C)
(instruction(icl>information)).

Noun candidate (No.)	Distance
(8)	**7**
(9)	11

processed. Only the verb "have" contains ambiguous UWs in the UNL Ontology:

(4) $\texttt{have}(\texttt{agt} > \texttt{person}, \texttt{obj} > \texttt{thing})$

(5) $\texttt{have}(\texttt{agt} > \texttt{thing}, \texttt{obj} > \texttt{thing})$

(6) $\texttt{John}(\texttt{iof} > \texttt{person})$

(7) $\texttt{list}(\texttt{icl} > \texttt{set})$

For space reasons, we cannot show the location of these UWs inside UNL Ontology. See Table III for the best candidate of the verb "have".

Next, Step 4 is applied because there is a noun-noun relation containing a word for which a best candidate has not been calculated ("instruction") and the related noun has a best candidate ("list"). The noun "instruction" has two UWs:

(8) $\texttt{instruction}(\texttt{icl} > \texttt{information})$

(9) $\texttt{instruction}(\texttt{icl} > \texttt{statement})$

Then, we use the Equation 4 to calculate the best candidate of "instruction", indicated in the Table IV.

IV. GRAPH-BASED WSD

A. Overview

The graph-based approach from [10] has been adapted to work with supervised data, since our approach uses the UNL Ontology as data source. Tools for this method are the same as those used by the method in Section III.

Table V shows the semantic measures used in the method. All formulas require information of two elements: Least Common Subsumer (LCS) and Information Content (IC).

1) Least Common Subsumer: The Least Common Subsumer (LCS) of concepts "$c_1, c_2, \ldots, c_n$" is the most specific concept subsuming "$c_1, c_2, \ldots, c_n$". For instance, as can be seen in Figure 2, LCS for concepts "John(iof>person)" and "Alice(iof>person)" is the concept "person(icl>human)"; and for concepts "John(iof>person)" and "car(icl>vehicle)" the LCS is the concept "thing".

Table V
SEMANTIC MEASURES USED IN THE GRAPH-BASED METHOD.

Formula	Source type	Semantic measure
Jiang & Conrath	Ontology + Corpus	Distance
Li et al.	Ontology + Corpus	Similarity
Lin	Ontology + Corpus	Similarity
Resnik	Ontology + Corpus	Similarity
Wu & Palmer	Ontology	Similarity

2) Information Content: The Information Content (IC) of a concept is obtained as follows:

$$IC(c) = -\log P(c) \tag{5}$$

where $P(c)$ is the probability of finding an instance of the concept c in a corpus.

Since the data source is an ontology, Equation 5 is not applicable. However, it is possible to use ontologies to calculate the Intrinsic Information Content of concepts, with the equation proposed in [9]:

$$IC(c) = 1 - \frac{\log hypo(c) + 1}{log(tc)} \tag{6}$$

where $hypo(c)$ means the number of hyponyms under a concept c and tc is a value that represents the total of concepts in the taxonomy. In consequence, equation 6 is used to calculate the IC in this method.

In principle, the Equation 6 was used with WordNet but, according to [9], other taxonomies would be tested as well. In this work, we use UNL Ontology instead.

The concepts in a taxonomy express their IC based on the amount of hyponyms they contain. Concepts with few or no hyponyms provide the most specific information in the taxonomy, while concepts with many hyponyms would require some differentiation at deeper levels. In such a way, concepts with few or no hyponyms would express more information than concepts with many hyponyms.

B. Method

The steps for this method are the following:

1) **Syntactic analysis:** Same as in Section III-C.
2) **Extraction of verb and noun candidates:** Same as in Section III-C.
3) **Calculate similarity or distance between word senses:** The following semantic measures are used for this purpose:
   - Jiang and Conrath [2]:

$$Sim_{JC}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(LCS)} \tag{7}$$

- Li et al. [3]:

$$Sim_{Li}(c_1, c_2) = \begin{cases} e^{-\alpha l} * \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if } c_1 \neq c_2 \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

- Lin [4]:

$$Sim_{Lin}(c_1, c_2) = \frac{2 * IC(LCS)}{IC(c_1) + IC(c_2)} \quad (9)$$

- Resnik [8]:

$$Sim_{R}(c_1, c_2) = IC(LCS) \quad (10)$$

- Wu and Palmer [12]:

$$Sim_{WP}(c_1, c_2) = \frac{2 * depth(LCS)}{depth(c_1) + depth(c_2)} \quad (11)$$

In all equations, c_1 and c_2 represent the concepts for which the similarity value will be calculated. Additionally, in the cases where it applies, LCS represents the least common subsumer of the two concepts; $depth$ refers to the depth of either a concept or the LCS; and IC is the information content of a concept or the LCS.

As for equation 8, l represents the shortest path between the two concepts; h is the level of the LCS in the tree; and parameters α and β are the contributions of l and h, respectively. According to [3] the optimal values are: $\alpha = 0.2$, and $\beta = 0.6$.

4) **Calculate score for each noun and verb candidate:** First, degree values for all nodes are calculated by applying the following In-Degree Centrality equation:

$$indegree(v) = \sum_{(u,v) \in E} w_{uv} \quad (12)$$

where v is the node for which the score is being calculated, and w_{uv} is the weight of the edge that connects the nodes u and v. Since the graph is undirected, there is no information of edges going to or coming from any node. Therefore, this equation will consider all edges connected to a node.

Next, the score for each node can be calculated as follows:

$$score(v) = \frac{indegree(v)}{max_{in}} \quad (13)$$

where max_{in} represents the maximum in-degree in the graph.

Finally, the nodes with the highest score for each word will be considered as the best candidates for word meanings.

V. Semi-automatic Conversion from NL to CDL

The problem of the conversion from NL to CDL is that some expressions from natural languages can be interpreted in several ways. Therefore, it is necessary to solve the ambiguities first.

One of the characteristics of the UNL Ontology is that it provides unambiguous data. Restricted UWs include a relation and a noun class that contains the given Restricted UW, but also there are cases where more than one restriction can be found. For instance, some UWs of verbs include two restrictions: one for words that play role as agents, and one for words that play role as objects. Logical constraints restrict the interpretation of a UW to a subset or to a specific concept.

The WSD method explained in Section III aims to avoid the presence of ambiguities in the text, in order to perform an accurate conversion to CDL. However, it is still not possible to make the conversion in a fully-automatic way. In consequence, a semi-automatic process would allow users to minimize the work load that takes to choose word meanings by selecting best candidates. If this is achieved, semantic computing based on CDL and its purposes can be accomplished more easily.

VI. CDL Graphical Editor

CDL Graphical Editor is a graphical user interface built for testing the WSD method, as well as conversion from natural language text into CDL. It employs all the tools mentioned in Section III-A and some others that will be mentioned hereafter. This application is composed of seven modules, whose workflow is shown in Figure 3:

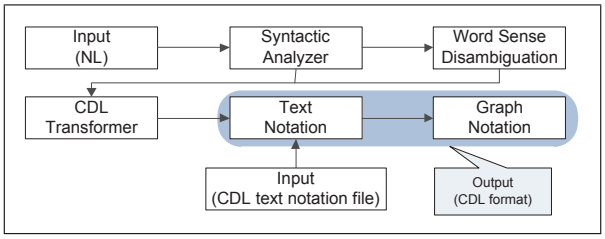

Figure 3. CDL Graphical Editor workflow.

Input: allows users to input sentences in English language. The input can be typed directly or imported from a text file.

Syntactic Analyzer: parses the input text, in order to obtain the sentence's dependency relations, tree structure, and words features such as lemma and part-of-speech tags.

Word Sense Disambiguation: enables users to select best candidates for word meanings. Candidates are taken from UNLKB (explained in Section III-A1).

CDL Transformer: analyzes the syntactic dependency relations that connect the words in natural language

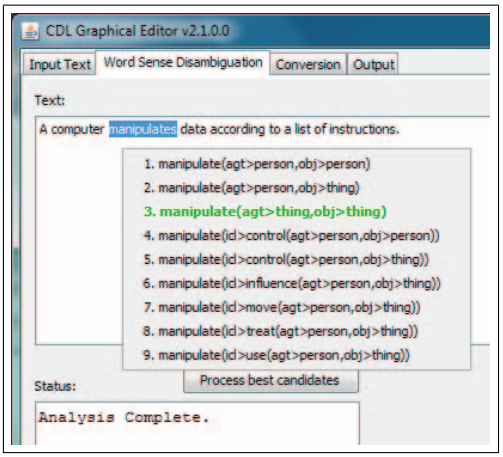

Figure 4. Best candidate selection in CDL Graphical Editor.

sentences. Next, by the use of a simple rule-based conversion method, transforms relations from syntactic to semantic.

Text Notation: displays the text notation of CDL structure received either from CDL Transformer module or by loading it from input file.

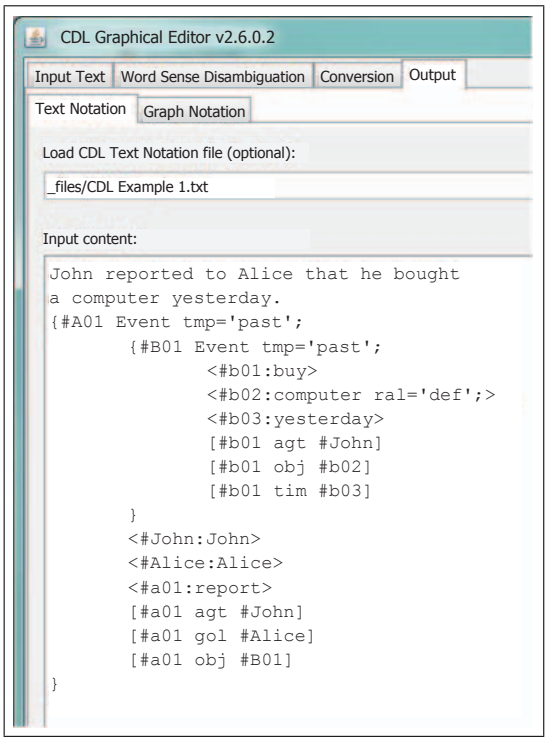

Figure 5. CDL Graphical Editor displaying CDL text notation.

Graph Notation: represents CDL in a network structure. For this goal, we use Prefuse Information Visualization Toolkit[9]. Prefuse takes nodes and edges information in two separate tables and builds the corresponding graph.

[9]http://www.prefuse.org/

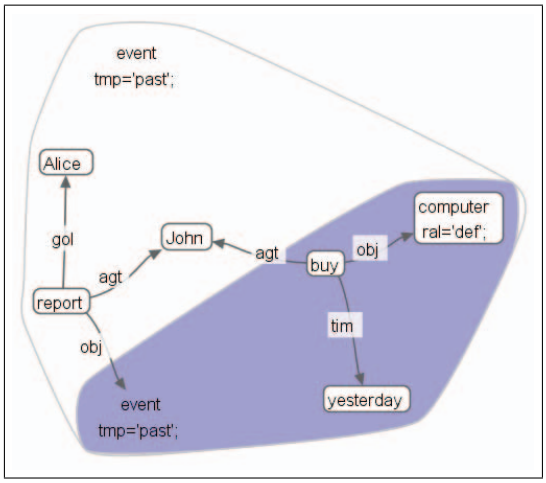

Figure 6. Graph notation of CDL generated by Prefuse toolkit.

Table VI
VOCABULARY BUILT FROM UNL ONTOLOGY.

Word Category	Total Concepts
Noun	4700
Verb	2676
Adjective	319
Adverb	320
Predicative	1481
TOTAL	**9496**

Input (CDL format): loads a text file that contains a CDL description, and transfers the contents to the Text Notation module.

VII. EXPERIMENTS AND RESULTS

For testing the WSD approaches, we have built a database that contains UWs from UNL Ontology (see Table VI). In this occasion, only with verbs and nouns were considered. Also, a set of sentences was extracted from the Encyclopedia of Life Support Systems (EOLSS)[10].

We present preliminary results after applying our method of analysis of semantic relations and the graph-based method to some test sentences. These results indicate in percentage the overall accuracy, that is, whether the methods could successfully determine the best candidates for word meanings. The formula for calculating the overall accuracy is as follows:

$$OAcc = \frac{\sum_{i=1}^{n} Acc_i}{n} \tag{14}$$

$$Acc_i = \frac{TCSs_i}{TAWs_i} \tag{15}$$

where Acc_i is the percentage of accuracy for sentence i, n represents the total of sentences, $TCSs_i$ is the total

[10]http://www.eolss.net/

Table VII
OVERALL ACCURACY FOR ALL METHODS.

Method		Overall Accuracy
Graph-Based	Jiang & Conrath	70.85%
	Li et al.	50.31%
	Lin	55.22%
	Resnik	60.52%
	Wu & Palmer	53.32%
Relations Analysis		66.28%

correct selections for sentence i, and $TAWs_i$ is the total of ambiguous words inside sentence i (based on word meanings available from UNL Ontology).

Selection of correct word meanings are based on human judgment. In consequence, best candidates are compared with those from human selection and the total of correct selections is increased for each positive case, as shown in the Algorithm 1:

Algorithm 1 TCS calculation

$TCS \leftarrow 0$
$HCS[] \leftarrow get\_human\_correct\_selections()$
$BCGroup[] \leftarrow get\_words\_best\_candidates()$
for each $BC \in BCGroup$ **do**
  if $BC \in HCS$ **then**
    $TCS \leftarrow TCS + 1$
  end if
end for

After applying Equation 14 for each method, we obtained the results displayed in the Table VII.

UNL Ontology is still a resource under growth, and concepts for some words are still not present. In order for sentences to be tested, their words and nouns must have meaning candidates available in the database of UWs. In consequence, from the set of sentences (160 in total) only 33 could be tested. The results from Table VII are based on this group of 33 sentences.

In spite of the obtained results, we consider that the method based on analysis of semantic relations fits better for the nature of the data source that was used due to the following reasons:

- It does not only rely on distance between concepts, but also considers syntactic and semantic relations for the best candidates calculation.
- Best candidates are constrained by a word-to-class relation with concepts from upper categories.
- It does not need the calculation of Information Content.

UNL provides functions to indicate which relations between two UWs are possible. We consider to analyze if the inclusion of these functions will improve the accuracy of the method of semantic relations analysis. However, the good performance of the method depends primarily on the structure of the ontology, that is, concepts should exist for every meaning of a word and all these concepts must be well organized in the hierarchy.

Once collected more data from UNL Ontology, we plan to make additional tests with a bigger set of sentences. There are some baselines to compare the accuracy of the methods [7]: Random Baseline and First Sense Baseline. The first takes a random choice of a sense for each word; the second selects a word sense based on its ranking, which is determined by the frequency of occurrence of the word sense occurs in a corpus. Since we are not working with corpus, we will compare accuracies with the random baseline in further tests.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presented a WSD approach based on selection of best candidates for semi-automatic conversion of NL text into CDL format. The approach analyzes semantic relations and noun classes from lexical units called UWs, and determines the best candidates by considering their total of valid relations and distances between concepts. This approach was compared with a graph-based method for WSD, which becomes another possible way to disambiguate words.

By the nature of the data available in UNL Ontology, the approach based on the analysis of semantic relations is better, due to the type of analysis it does with respect to the elements contained in UWs.

The experiments in this work produced some preliminary results that we intend to extend as long as more data becomes available in the UNL Ontology. Results indicate that the existence of a proper correspondence of syntactic and semantic relations may contribute to a disambiguation with high precision. Moreover, it is necessary that the source of data contains the adequate concepts defined in its structure.

As future work, it has been considered to include analysis of statistical data, in order to improve the performance of the WSD approach. Most of the coming tasks will be focused on this goal.

REFERENCES

[1] E. Agirre and P. Edmonds (eds.), *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[2] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proceedings of the International Conference on Research in Computational Linguistics*, 1997.

[3] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *Journal of the IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[4] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, 1998, pp. 296–304.

[5] P. Moreda, M. Palomar, and A. Suárez, "Assignment of Semantic Roles Based on Word Sense Disambiguation," *Advances in Artificial Intelligence*, vol. 3315, pp. 256–265, 2004.

[6] P. Moreda and M. Palomar, "The Role of Verb Sense Disambiguation in Semantic Role Labeling," *Advances in Natural Language Processing*, vol. 4139, pp. 684–695, 2006.

[7] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[8] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *International Joint Conference on Artificial Intelligence*, vol. 14, pp. 448–453, 1995.

[9] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," in *Proceedings of ECAI 2004*, Valencia, Spain, 2004, pp. 1089–1090.

[10] R. Sinha and R. Mihalcea, "Unsupervised graph-based word sense disambiguation using measures of word semantic similarity," in *Proceedings of the IEEE International Conference on Semantic Computing*, 2007, pp. 363–369.

[11] H. Uchida, M. Zhu, and T. D. Senta, *The Universal Networking Language*. UNDL Foundation, 2005.

[12] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, 1994, pp. 133–138.

[13] T. Yokoi, H. Yasuhara, H. Uchida, M. Zhu, and K. Hasida, "CDL (Concept Description Language): A Common Language for Semantic Computing," in *WWW2005 Workshop on the Semantic Computing Initiative (SeC2005)*, Makuhari, Japan, 2005.