Journal of Mobile, Embedded and Distributed Systems, vol. V, no. 3, 2013
ISSN 2067 – 4074

WSD Implementation for Processing Improvement of Structured Documents

Madalina ZURINI
Bucharest University of Economic Studies
ROMANIA
madalina.zurini@gmail.com

Abstract. The term of word sense disambiguation, WSD, is introduced in the context of text document processing. A knowledge based approach is conducted using WordNet lexical ontology, describing its structure and components used for the process of identification of context related senses of each polysemy words. The principal distance measures using the graph associated to WordNet are presented, analyzing their advantages and disadvantages. A general model for aggregation of distances and probabilities is proposed and implemented in an application in order to detect the context senses of each word. For the non-existing words from WordNet, a similarity measure is used based on probabilities of co-occurrences.

Key-Words: WordNet, supervised classification, word similarity, context similarity, ontology.

1. Introduction

For the acquisition of knowledge in artificial intelligence, two approaches defined in [1] are used:

- *transfer process between human to knowledge base,* process with a major disadvantage given by the fact that the one who has knowledge cannot easily identify it;
- *conceptual modeling process* by building models in which are placed the new knowledge as they are acquired, this process leading to the appearance of the ontology as a systematic organization of knowledge, data of the reality, leading to the construction of theories upon what it exists.

An essential role of ontology is to be reused in multiple applications. Mapping two or more ontologies is called alignment. This task is particularly difficult, the main cause of limitation in extending existing ontologies [1].
Direction that follows the ontology is supported by the introduction of artificial intelligence techniques to emulate the mental representation of concepts used, and the interpenetration of these links.
The kernel of the ontology is defined as a system $\mathcal{O} = (\mathcal{L}, \mathcal{F}, \mathcal{C}^*, \mathcal{H}, ROOT)$, where:

- $\mathcal{L}$ is the lexicon formed out of the terms from the natural language;
- $\mathcal{C}^*$ a set of concepts;
- $\mathcal{F}$ represents the reference function that maps the set of terms of the lexicon to the set of concepts;
- $\mathcal{H}$ is the hierarchy of the taxonomy given by the direct, acyclic, transitive and reflexive relation;
- $ROOT$ is the starting point upon which the hierarchy is built on.

There are two types of ontologies as defined in [1], depending on the area in which they are used:

- ontologies for knowledge-based systems are characterized by a relatively small number of concepts, but linked by a large and varied relationships, concepts are grouped into complex conceptual schemes or scenarios and for each concept there can be one or more customizations;
- lexicalized ontologies, including a large number of concepts linked by a small number of relationships, like WordNet ontology concepts that are represented by sets of synonymous words, these ontologies are used in human language processing systems.

The process of WSD is added in the general analysis of the documents in order to obtain an accurate result of the implementation, analysis also done in the papers [3], [4], [5] and [6].
Section 2 contains the presentation of WordNet ontology along with the

components and relations, resulting in a graph representation.
Word Sense Disambiguation algorithm is presented in chapter 3, where a mathematic view is highlighted.
In chapter 4, the framework of implementation of the WSD algorithm is integrated in the analysis, including code source, application snapshots with results and interpretation. The paper ends with the conclusions and future work.

2. Graph representation using WordNet components

Tree representation of the links between concepts is based on the WordNet ontology tree creating a form of words/synsets represented by nodes and links, arcs, represented by types of WordNet semantic relations between concepts, [2]. Top-bottom representation consists of a root, the point at which splits all existing links between concepts, which is called the root entity.
For the concept *car* in the WordNet ontology there are five ways identified with description and structure to the existing synset for each sense individually, Figure 1, using WordNet 2.1 Browser.

The noun car has 5 senses (first 3 from tagged texts)

1. (598) **car**, auto, automobile, machine, motorcar -- (a motor vehicle with four wheels; usually propelled by an internal combustion engine; "he needs a car to get to work")
2. (24) **car**, railcar, railway car, railroad car -- (a wheeled vehicle adapted to the rails of railroad; "three cars had jumped the rails")
3. (1) cable car, **car** -- (a conveyance for passengers or freight on a cable railway; "they took a cable car to the top of the mountain")
4. **car**, gondola -- (the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant)
5. **car**, elevator car -- (where passengers ride up and down; "the car was on the top floor")

Figure 1 – Senses of car noun from WordNet ontology

Each sense becomes leaf node for the semantic graph representation using semantic relations.
Based on the relations of *"is-a"* type, the graph representation is formed, figure 2.

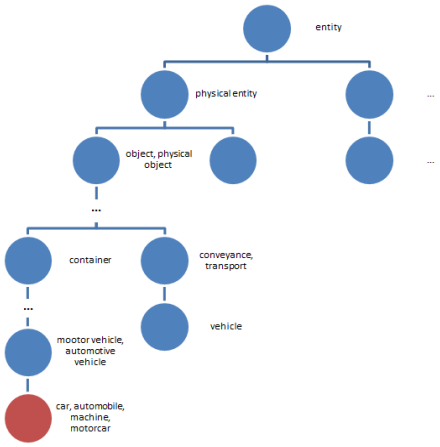

Figure 2 – Graph associated to the first sense of car noun using is-a relations

Synonymy relations are shown in a graphical representation in the same synset node type, node that contains all the concepts synonymous. Hyponymy and hypernymy relations type levels build the graph starting from the root, the concept of "entity", and reaching the leaf represented by the noun. On the basis of representation through a graph, one can evaluate the similarity or correlation between two concepts in the WordNet ontology. This metric is then used in the evaluation of applications for text documents, as well as supervised classification and clustering, the semantic problem solving.
The similarity between the two concepts in the is-a hierarchy of the graph associated to the ontology WordNet quantifies how much resemble those objects based on information held on schedule [8]. Measurement correlation and the distance between words is used in applications such as identifying contextual meanings of words, determining the structure of text documents, creating automatic summaries, information extraction and automatic indexing [9].
For understating the way of similarity calculation between the WordNet concepts, the graph associated to the WordNet ontology is given as starting point. Figure 3 contains part of the representation for the examples car and bicycle.

ISSN 2067 – 4074

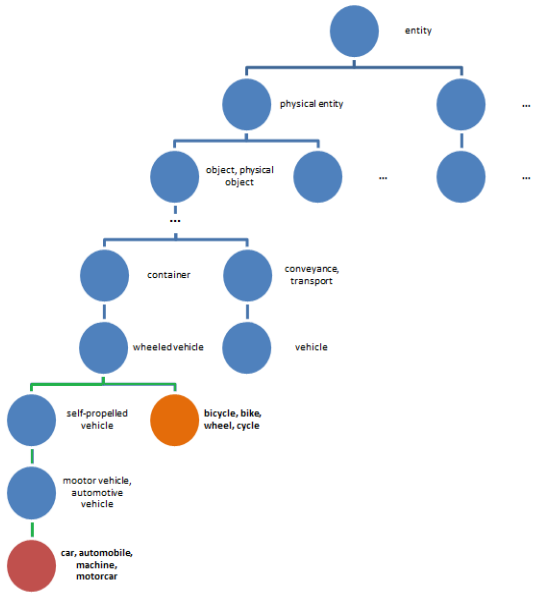

Figure 3.Graph associated to car and bicycle nouns using is-a relations

In the context of similarity identification between c_1 and c_2 concepts with multiple senses, the metric result is given by the maximum between the values of the similarity metric of each senses of concepts c_1 and c_2.

3. WSD Algorithm

Automatic evaluation of contextual meanings of words had an interest and concern since the beginning of natural language processing. Evaluation meaning is not seen as an independent business, but as an intermediate step and necessary in order to achieve the semantic processing of text objects, [16].
One way to solve the problem of choosing the contextual meaning of a word in the context in which the word is polysemy is to extend analysis at word level way, increasing the size of the representation of text documents directly proportional to the number of senses added in the analysis, and training base to be able to perform statistical analysis of the occurrence of contextual meanings, and correlations with other words that deal directly.
The kernel of the sense disambiguation algorithm consists in computing the semantic similarity using the taxonomy of WordNet ontology, [18].
The general model for describing the problem of context sense choosing is given by the existence of a set of key words from which a phrase is formed of:

$$F = \{w_1, w_2, \dots, w_f\}$$

where:

- F is the analyzed phrase;
- f is the number of words;
- w_i is the i word of the phrase F.

For each word from F phrase, the set of senses is formed:

$$s_i = \{s_{i1}, s_{i2}, \dots, s_{Card(s_i)}\}$$

where s_{ij} is the j sense of word w_i.
Let w_p be a polysemy word, the problem of sense identification summarizes in choosing the sense that has maximum similarity between the word and each other word found in the phrase. Using the similarity measures previously described, the maximization model is:

$$w_p = \left\{ s_{pj} \middle| j = arg \max_{j=1,Card(s_p)} \frac{\sum_{\substack{k=1 \\ k \neq p}}^{f} d_{SIM}(w_{pj}, w_k)}{f-1} \right\}$$

If more polysemy words exist, the algorithm is repeated for each one, resulting in a sense that maximizes the context similarity. Each polysemy word is analyzed according to the words and senses found after it. Comparing the probabilities for the first polysemy word with the rest, the resulted sense maximizes the semantic information.

4. Implementation of WSD

Different implementations of Word Sense Disambiguation process were proposed in the papers [17] and [19]. For the evaluation of contextual senses for polysemy words, similarity metrics are used. Figure 4 contains the section from the application that calculates Path Length and Wu & Palmer metrics.

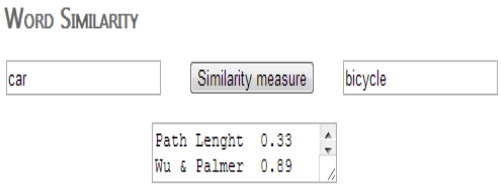

Figure 4. Similarity evaluation of two words

```
WordSimilarity wordSim = new WordSimilarity();
MyWordInfo wordinfo1 = new MyWordInfo(word1, PartsOfSpeech.Noun);
HierarchicalWordData hwd1 = new HierarchicalWordData(wordinfo1);
MyWordInfo wordinfo2 = new MyWordInfo(word2, PartsOfSpeech.Noun);
HierarchicalWordData hwd2 = new HierarchicalWordData(wordinfo2);
HierarchicalWordData[] hwd = { hwd1, hwd2 };
ancestor = wordSim.FindLeastCommonAncestor(hwd, out distance, out lcaDepth, out depth1, out depth2);

float dist1 =wordSim.GetSimilarity(hwd1, hwd2, 1);
float dist2 = wordSim.GetSimilarity(hwd1, hwd2, 2);
txt_results.Text = "Path Lenght  " + dist1.ToString() + "\r\n";
txt_results.Text += "Wu & Palmer  " + dist2.ToString() + "\r\n";
```

Figure 5. Source code for two words similarity measure

Table 1 – Formulas for similarity measure

Similarity metric	Calculation formula	Variables used
Path Length	$d_{PATH}(c_1, c_2) = \frac{1}{lg(c_1, c_2)}$	$lg(c_1, c_2)$ is the minimum length of the path between c_1 and c_2 nodes.
Wu & Palmer	$d_{WP}(c_1, c_2) = \frac{2 \cdot lg(l(c_1, c_2))}{lg(c_1, l(c_1, c_2)) + lg(c_2, l(c_1, c_2)) + lg(c_1, l(c_1, c_2))}$	$l(c_1, c_2)$ is the first mutual parent of c_1 and c_2 nodes.

Figure 5 contains the source code of the similarity between two words, word1 and word2, using the hierarchies associated to the terms, hwd1 and hwd2.
The calculation formulas of the similarity metrics between c1 and c2 words are presented.

Table 1 contains the formulas referring to Path Lenght and Wu & Palmer similarity metrics, described in [11], [12], [13], [14] and [15].

```
public float GetSimilarity(HierarchicalWordData word1, HierarchicalWordData word2, int strategy)
{
    if (word1.WordInfo.Pos != word2.WordInfo.Pos || word1.WordInfo.Pos == PartsOfSpeech.Unknown) return 0.0F;
    if (word1.WordInfo.Word == word2.WordInfo.Word) return 1.0F;

    int pathLength, lcaDepth, depth_1, depth_2;
    FindLeastCommonAncestor(new HierarchicalWordData[2] { word1, word2 },
        out pathLength, out lcaDepth, out depth_1, out depth_2);

    if (pathLength == int.MaxValue) return 0.0F;
    float sim = 0.0F;
    if (strategy == 1)//Path Length
    {

        if (pathLength == 0) return 1.0F;
        else
            sim = 1.0F / (float)pathLength;
    }
    else
        if (strategy == 2) //Wu & Palmer
        {
            if (pathLength == 0) return 1.0F;
            else
                sim = (float)(lcaDepth) / (float)(depth_1 + depth_2);
        }

    return (float)Math.Round(sim, 2);
}
```

Figure 6. Source code for Path_Lenght and Wu & Palmer similarity measures

ISSN 2067 – 4074

Figure 6 contains the source code for the calculation of the similarity metrics Path Length and Wu & Palmer using the hierarchical representation of the words within the graph associated to the WordNet ontology and the nodes' densities.
Figure 7 contains the frame in which the set of words can be inserted along with the results of the contextual semantic analysis using the proposed algorithm defined previously. For each word, the result is the most probable sense in terms of similarity maximization among the concepts of the phrase.

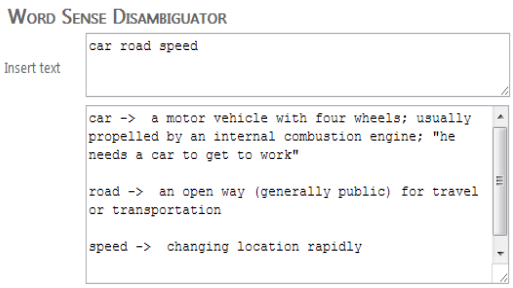

Figure 7 – Context senses for the words inside the phrase

The results obtained for the phrase containing the words *car* and *bicycle* are presented in figure 8.

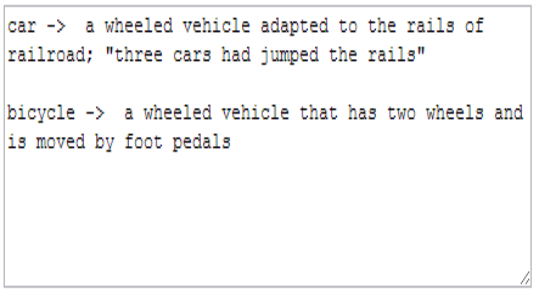

Figure 8. Contextual senses for car and bicycle

An additional possibility is to generate all the senses of the words from which the phrase is formed of, figure 9, in order to evaluate the correctness of the contextual senses calculation.

Nr	Synset	Description
1	car auto automobile machine motorcar	a motor vehicle with four wheels; usually propelled by an internal combustion engine; "he needs a car to get to work"
2	car railcar railway_car railroad_car	a wheeled vehicle adapted to the rails of railroad; "three cars had jumped the rails"
3	cable_car car	a conveyance for passengers or freight on a cable railway; "they took a cable car to the top of the mountain"
4	car gondola	the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant
5	car elevator_car	where passengers ride up and down; "the car was on the top floor"
1	bicycle bike wheel cycle	a wheeled vehicle that has two wheels and is moved by foot pedals

Figure 9. All senses for the words in the analyzed phrase

In figure 10, it is presented the source code of the WSD process.

```
txt_results.Visible = true;
txt_results.Text = "";
words = txt_words.Text.ToString().Split(' ');

WNCommon.path = @"C:\Program Files\WordNet\2.1\dict\";

MyWordInfo[] words_info = new MyWordInfo[words.Length];
for (int i = 0; i < words_info.Length; i++)
{
    words_info[i] = new MyWordInfo(words[i], Wnlib.PartsOfSpeech.Noun);
}
WordSenseDisambiguator WSD = new WordSenseDisambiguator();
MyWordInfo[] info = WSD.Disambiguate(words_info);
```

Figure 10. Source code for WSD process

The testing process consists in running a set of phrases priory contextual sense classified. The metric used for evaluating the WSD correctness, IC_{WSD}, is defined using:

$$IC_{WSD} = \frac{\sum_{i=1}^{nr\_wsd} w_i}{nr\_wsd} \times 100$$

where:

- *nr_wsd* is the number of polysemy words existing in the phrases used for testing;
- w_i represents the association between the priory sense of the i word with the sense generated by WSD algorithm, based on the formula:
- $w_i = \begin{cases} 1, sens\_aprioric_i = sensWSD_i \\ 0, otherwise \end{cases}$
- $sens\_aprioric_i$ is the priory sense associated to the word *i*;
- $sensWSD_i$ is the sense generated by WSD algorithm for the *i* word.

A testing set formed out of 100 phrases is used, containing *nr_wsd=200* polysemy words. After running WSD algorithm, the value of IC_{WSD} indicator is 94%.

5. Conclusion and future work

Adding a context analysis for the words that has multiple meaning according to the neighbor words increases the performance of text document processing and representation. The proposed aggregation method of the probabilities of each sense of the existing words within a phrase optimizes the correctness of the word sense disambiguation process, taking into account the space and time consuming elements.
WordNet ontology is added as an external knowledge base used for an up level representation of English concepts, resolving the problem of similarity among the existing concepts.
The results of the WSD process indicate a correctness level of 94% for the testing set used.

Acknowledgments

This work was cofinaced from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards".

References

[1] Trausan-Matu, S. "Inteligenta artificiala", 2004, Available online at: http://www.racai.ro/~trausan/ia.pdf
[2] WordNet. A lexical database for English, Available online at: http://wordnet.princeton.edu/wordnet/related-projects/
[3] Hessami, E., Mahmoudi, F., Jadidinejad, H. „Unsupervised Graph-based Word Sense Disambiguation Using lexical relation of WordNet", International Journal of Computer Science Issues, Vol. 8, Nr. 3, 2011, pg. 225-230, ISSN 1694-0814
[4] WordNet Statistics: Available online at: http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
[5] Gonzalez, A., Rigau, G., Castillo, M. „A graph-based method to improve WordNet Domains", Proceeding CICLing'12 Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing, Vol 1, 2012, pg. 17-28, ISBN 978-3-642-28603-2
[6] Elberrichi, Z., Rahmoun, A., Bentaalah, M.A. „Using WordNet for Text Categorization", The International Arab Journal of Information Technology, Vol. 5, Nr. 1, 2008, pg. 16-24, ISSN 1683-3198
[7] Passos, A., Wainer, J. „Wordnet-based metrics do not seem to help docucment clustering", 2009, Available online at: http://www.ic.unicamp.br/~tachard/docs/wncluster.pdf
[8] Pedersen, T., Patwardhan, S., Michelizzi, J. „WordNet::Similarity – Measuring the Relatedness of Concepts", Proceeding HLT-NAACL--Demonstrations '04 Demonstration Papers at HLT-NAACL, May, 2004, Boston, pg. 38-41
[9] Budanitsky, A., Hirst, G. „Evaluating WordNet-based Measures of Lexical Semantic Relatedness", Journal Computational Linguistics, Vol. 32. Nr. 1, 2006, pg. 13-47, ISSN 2180-1266
[10] Peng, Q., Zhao, L., Yu, Y., Fang, W. „A New Measure of Word Semantic Similarity based on WordNet Hierarchy and DAG Theory", International Conference on Web Information Systems and Mining, 2009, pg. 181-185, ISBN 978-0-7695-3817-4
[11] Blanchard, E., Harzallah, M., Briand, H., Kuntz, P. "A typology of ontology-based semantic measures", Proceeding of EMOI-INTEROP 05, Portugal, June 2005
[12] Buhanitzky, A., Hirst, G. "Evaluating WordNet-based Measures of Lexical Semantic Relatedness", Journal Computational Linguistics, Vol. 32, Nr. 1, 2006, pg. 13-47, ISSN 1530-9312
[13] Yang, D., Powers, D.M.W. „Measuring Semantic Similarity in the Taxonomy of WordNet", 28th Australasian Computer Science Conference, Newcastle, Australia, 2005, pg. 315-322
[14] Lewis, W.D. "Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring Semantic Similarity", Language in Cognitive Science, 2001, pg. 9-16, Available online at: http://coyotepapers.sbs.arizona.edu/CPXII/Lewis.pdf
[15] Richardson, R., Smeaton, A., Murphy, J. „Using WordNet as a Knowledge Base for Measuring Semantic Similarity between Word", Technical Report, Working paper CA-1294, School of Computer Applications, Dublin City University, 1994

ISSN 2067 – 4074

[16] Kamali, S. „Some Experiments in Word Sense Disambiguation", 2001, Available online at: https://cs.uwaterloo.ca/~s3kamali/courses/word-sense-disambiguation.pdf
[17] Xiaobin, L., Szpakowicz, S., Matwin, S. „A WordNet-based Algorithm for Word Sense Disambiguation", Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pg. 1368—1374
[18] Resnik, P. „Disambiguating Noun Grouping with Respect to WordNet Senses", Natural Language Processing Using Very Large Corpora Text, Speech and Language Technology, Vol. 11, 1999, pg. 77-98, ISBN 978-90-481-5349-7
[19] Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R. „Adding Dense, Weighted Connections to WordNet", 2005, Available online at: https://wordnet.princeton.edu/wordnet/publications/jbj-jejufellbaum.pdf