Building Semantic Corpus from WordNet

Lubomir Stanchev
Indiana University - Purdue University Fort Wayne
Fort Wayne, IN, USA
e-mail: stanchel@ipfw.edu

***Abstract*—We propose a novel methodology for extracting semantic similarity knowledge from semi-structured sources, such as WordNet. Unlike existing approaches that only explore the structured information (e.g., the hypernym relationship in WordNet), we present a framework that allows us to utilize all available information, including natural language descriptions. Our approach constructs a *semantic corpus*. It is represented using a graph that models the relationship between phrases using numbers. The data in the semantic corpus can be used to measure the similarity between phrases, the similarity between documents, or to perform a semantic search in a set of documents that uses the meaning of words and phrases (i.e., search that is not keyword-based).**

I. Introduction

Most contemporary search engines return resources that contain keywords from the input query in their description or content. For example, a search for “The Big Apple” may not return resources that are associated with “New York” because the search engine lacks information about the relationship between the two concepts. Similarly, a drug information system may skip articles on ascorbic acid when the user requests information about Vitamin C. In both cases, the information system may lack the information about the semantic similarity between different concepts. In this paper, we address the problem of modeling the structure and computing the content of a semantic corpus that stores such information.

The problem of evaluating the semantic relationship between natural language expressions, or *phrases*, is intrinsically hard because computers are not as proficient as humans in understanding natural language text. Although significant effort has been put in automated natural language processing (e.g. [5], [6], [15]), current approaches fall short of understanding the precise meaning of human text. In fact, the question of whether computers will ever become as fluent as humans in understanding natural language text is an open problem. In this paper, unlike most natural language processing applications, we do not parse text and breakdown sentences into the primitive elements of the language. Instead, we use freely-accessible resources, such as WordNet, to collect evidence about the strength of the relationship between different phrases.

Current approaches that extract information about phrase similarity from freely-accessible sources focus on the structured information. In particular, most papers that deal with WordNet (e.g. [13], [27]) adapt the approach taken in [20] that semantic similarity can be measured solely based on the inheritance (a.k.a. is-a) links and possibly data about the specificity of the phrases (i.e., their information content - see [19], [14], [10]). More recent papers, such as [28], explore additional relationship between words, such as the holonym (a.k.a. part-of) relationship. Although these approaches work well in practice and produce similarity data that closely correlates to data from human studies, such as [16], we show that there is room for improvement. In particular, unstructured information, such as the definition of a phrase or an example use of a phrase, is not considered. For example, the WordNet definition of one of the senses of “New York” is that it is a city that is located on the Hudson river. This close relationship between “New York” and “Hudson river” is not considered by the algorithms of the papers that are cited in this paragraph because these algorithms do not process unstructured information.

In this paper, we propose a novel mechanism for measuring the semantic similarity between phrases and constructing a *semantic corpus* based on open-source resources that contain high-quality structured, semi-structured, and unstructured knowledge. We show how information from WordNet can be used to create a *phrase graph*, where the algorithm can be easily modified to include other sources (e.g., Wikipedia). The graph is created using probability theory and corresponds to a simplified version of a Bayesian network. The weights of the edges represent the probability that two phrases that are connected by an edge are related based on the available evidence, where the weight function is asymmetric. We experimentally validate the quality of our algorithm on two independent benchmarks: Miller and Charles ([16]) and WordSimilarity-353 ([4]). Our approach outperforms most existing algorithms because we process more information as input, including natural language descriptions, and we are able to apply this information to build a better model of the semantic relationships between phrases. The reader is encouraged to try our system, which is located at: http://softbase.ipfw.edu:8080/Similarity.

In what follows, in Section 2 we review related research. The major contributions of the paper are the introduction of the semantic corpus, see Section 3, and the introduction of two novel algorithms for measuring the semantic similarity between phrases, which are presented in Section 4. Section 5 shows how our system compares with existing systems that measure phrase similarity, while concluding remarks and areas for future research are outlined in Section 6.

II. Related Research

Existing research that applies Bayesian networks to ontologies deals with the uncertain or probabilistic information in the knowledgebase (e.g., [21], [18]). In this paper, we take a different approach and we do not use Bayesian networks to model uncertain information. In contrast, we create a probabilistic graph that stores information about the similarity of the different phrases. Unlike Bayesian networks, we store only the probability that a phrase is relevant given that an adjacent (in the graph) phrase is relevant (e.g., unlike Bayesian networks, we do not store the probability that a phrase is unrelated given that an adjacent phrase is unrelated).

The idea of creating a graph that stores the degree of semantic relationship between concepts is not new. For example, [11], [22] show how to create a graph that only represents inheritance of concepts, while [9] approximates the similarity of concepts based on information about the structure of the graph in which they appear. These papers, however, differ from our proposal because we suggest representing available evidence from all type of sources, including natural language description. Note that our proposal is different than the use of a semantic network ([25]) because the latter does not consider the strength of the relationship between the nodes in the graph.

Lastly, there are alternative methods to measure the semantic similarity between phrases. The most notable approach is the Google approach ([3]) in which the similarity between two phrases is measured as a a function of the number of Google results that are returned by each phrase individually and the two phrases combined. Other approaches that rely on data from the Internet include [1] and [12]. Although these approach produce good measurement of similarity, they have their limitations. First, they do not make use of structured information, such as the hyponym relationship in WordNet. Second, they do not provide evidence about how the two input phrases are related. In contrast, our approach can show the paths in the phrase graph between the search phrases, which serves as evidence that supports the similarity score.

III. Semantic Corpus

The semantic corpus consists of a *phrase graph*. The phrase graph is constructed from the information in WordNet 3.0 ([17]). The knowledgebase has information about approximately 150,000 different words. Although most entries in WordNet are single words, common phrases, such as "sports utility vehicle", are also present. We will use the term "word form" to refer to both words and word phrases. A set of *senses* is associated with every word form. For example, the senses "a seat for one person ..." and "the position of a professor", are two of the senses of the word chair. In the phrase graph, we create a node for every word form and every sense in WordNet. The label of a word form node is the actual word form, while the label of a sense node is the definition of the sense. Before describing how the edges in the phrase graph are constructed, we define the meaning of their weights.

Definition 1 (weighted edge)**.** *An edge in the phrase graph between nodes n_1 and n_2 with weight p describes that the probability that a user is interested in n_2 given that they are interested in n_1 is equal to p.*

A. Representing Textual Knowledge

Going back to our example word "chair", we will create edges between the node for the word and the nodes for the three different senses of the word. WordNet gives us the information about the frequency of use of each sense. The frequency of the first use is 35, the frequency of the second use is 2, and the frequency of the third use is 1. We will therefore create the outgoing edges for the node "chair" that are shown in Figure 1. The reasons is that, based on the available information, the probability that a user that requests information about the word "chair" is interested in the first sense of the word is equal to $35/38 = 0.92$. We assume that the information in WordNet tells us that 92% of the time when someone refers to a chair, they have in mind the first meaning (this conforms with Definition 1). In general, we set the weight of each edge to the frequency of the sense divided by the sum of the frequencies of all senses.

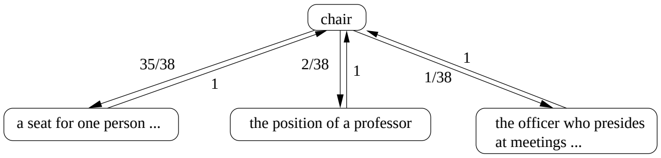

Fig. 1. Example Edges Between a Word and its Senses

According to WordNet, the third sense of the word "chair": "the officer who presides at the meetings ..." can be represented by either of the four words: "president", "chairman", "chairwomen", and "chairperson". We will draw an edge between every sense and its words, where all edges will have weight 1 (see Figure 1). This follows Definition 1 because the probability that someone is interested in a word given that they are interested in one of its senses is equal to 1. This is the case because all the words represent the same sense. In general, we will draw an edge from every sense to the words that represent it and assign weight of 1 to each edge.

Next, consider the second sense of the word "chair": "the position of a professor". "the", "of", and "a" are noise words. We will therefore create an edge between the node for the sense and the words "position" and "professor". Since the two words occur with the same frequency in the definition of the sense, the weight of the two edges will be both $0.5 * \alpha_1$, where α_1 is a parameter that tells us how likely it is that a user that is interested in a sense of a word form is also interested in one of the word forms in the definition of the sense. Figure 2 shows the portion of the graph that we described (for now, ignore the edge from the node "position"). We adopt this approach because, based on the available evidence, it is equally likely that a user will be interested in information about each of the two non-noise words in the definition of the sense. It is reasonable to assume that the probability that a user is interested in a word will increase if the word appears multiple times in the definition of the sense. In general, the weight of an edge between a sense of a word form and a word form in the definition of the sense will be computed as the number of occurrences of the word in the definition of the sense multiplied by α_1 and divided by the total number of non-noise words in the definition. Note that, if the definition of a sense contains a word form that is present in the graph, then we will draw an edge between the sense and the word form rather than drawing edges from the sense to all the words in the word form. For example, since "United States" appears in the definition of the only sense of the word "Mississippi", we will draw an edge from the node for the sense of the word "Mississippi" to the node "United States" and not to the two nodes: "United" and "States". In our system, we set $\alpha_1 = 0.5$. The reason is that non-noise words in the definition of a sense are related to the sense, but there is not necessarily a strong relationship.

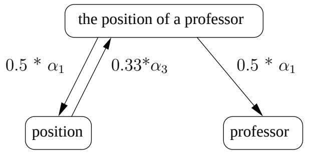

Fig. 2. Example Edges Between a Sense and the Words in its Definition

WordNet also includes example use for each word sense. For example, it contains the sentence "he put his coat over the back of the chair and sat down" as an example use of the first sense of word "chair". Let the probability that a user that is interested in this sense is also interested in one of the word forms in the example sentence be equal to α_2. Since an example use does not have as strong a correlation as the definition of a sense, it will be the case that $\alpha_2 < \alpha_1$. In our system, we set $\alpha_2 = 0.2$. Figure 3 shows the graph that is created for the example use of the first sense of the word *chair*. Note that the noise words have been omitted. In general, the weight of an edge between a sense of a word form and the word forms in its example use is computed as the number of occurrences of the word form in the example use multiplied by α_2 and divided by the total number of non-noise word forms in the example use.

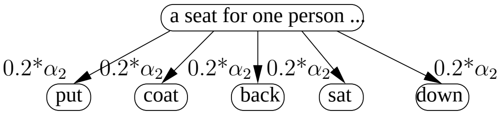

Fig. 3. Example Edges Between a Word Sense and the Words in its Example use

We will also draw edges from each word form to all nodes that contain the word form in their label. Let α_3 be the probability that someone who is interest in a word form is also interested in a word form that contains it in its definition or example use. The weight of an edge will then be equal to α_3 multiplied by the number of occurrences of the word form in the label divided by the total number of occurrences of the word form in the labels of all nodes. For example, if the word "position" occurred as part of the label of only three nodes and exactly once in each label, then there will be an edge between the nodes "position" and "the position of a professor" in Figure 2 with weight that is equal to $(1/3) * \alpha_3 = 0.33 * \alpha_3$ (see Figure 2). In our system, we set $\alpha_3 = 0.8$.

B. Representing Semi-Structured Knowledge

So far, we have shown how to extract information from textual sources, such as the text for the definition and example use of a word sense. We next show how semi-structured knowledge, such as the hyponym (a.k.a. is-a) relationship between senses, can be represented in the semantic corpus. Most existing approaches (e.g., [19]) explore these relationships by evaluating the *information content* of different word forms. Here, we adjust this approach and focus on the frequency of use of each word in the English language as described in the University of Oxford's British National Corpus ([2]).

Definition 2 (size of a word sense)**.** *Consider a word sense m. Let $\{w_i\}_{i=1}^n$ be the word forms for that sense. We will use $BNC(w)$ to denote the frequency of the word form w in the British National Corpus. Let $p_m(w)$ be the frequency of use the sense m, as specified in WordNet, divided by the sum of the frequencies of use of all senses of w (also as defined in WordNet). Then we define the* size *of m, denoted as $|m|$, to be equal to $\sum_{i=1}^{n}(BNC(w_i) * p_m(w_i))$.*

The size of a sense approximates its popularity. For example, according to WordNet the word "president" has six different senses with frequencies: 14, 5, 5, 3, 3 and 1. Let us refer to the fourth sense: "The officer who presides at the meetings ..." as m. According to Definition 2, $p_m(\mathit{president}) = 3/31 = 0.096$ because the frequency of m is 3 and the sum of all the frequencies is 31. Since the British National Corpus gives the word "president" a frequency of 9781, the contribution of the word "president" to the size of the sense m will be equal to $\mathit{BNC}(\mathit{president}) * p_m(\mathit{president}) = 9781 * 0.096 = 938.976$. Other words that represent the sense m will also contribute to the size of the sense.

In WordNet, X is a *hyponym* of Y if every X is a type of Y (e.g., "dog" is a hyponym of "canine"). Consider the first sense of the word "chair": "a seat for one person ...". WordNet defines 15 hyponyms for this sense, including senses for the words "armchair", "wheelchair", and so on. In the phrase graph, we will draw an edge between this first sense of the word "chair" and each of the hyponyms. Let the probability that someone that is interested in a sense is also interested in one of the sub-senses be equal to α_4. Since the hyponym relationship is structured information that comes directly from WordNet and expresses strong relationship, we set $\alpha_4 = 0.9$. In order to determine the weight of the edges, we need to compute the size of each sense. In the British National Corpus, the frequency of "armchair" is 657 and the frequency of "wheelchair" is 551. Since both senses are associated with a single word form, we do not need to consider the frequency of use of each sense. If "armchair" and "wheelchair" were the only hyponyms of the sense "a seat for one person ...", then the corresponding part of the phrase graph will look as shown in Figure 4. In general, the weight of each edge from n_1 to n_2 in this case is equal to α_4 multiplied by the size of n_2 and divided by the sum of the sizes of all the senses.

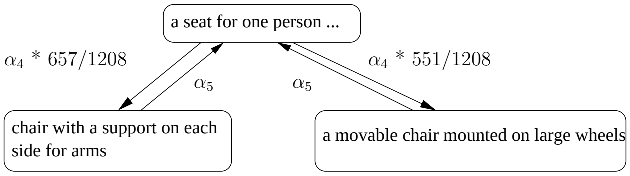
a seat for one person ...
α_4 * 657/1208
α_5
α_5
α_4 * 551/1208
chair with a support on each side for arms
a movable chair mounted on large wheels

Fig. 4. Example Edges Between a Word Sense and its Hyponyms

We will also draw edges for the *hypernym* relationship (the inverse of the hyponym relationship). Y is a hypernym of X if every X is a kind of Y (e.g., "canine" is a hypernym of "dog"). The weight of each edge will be the same and equal to the parameter α_5. For example, if a user is interested in the word sense "wheelchair", then they may be also interested in the resources that are associated with the first sense of the word chair. However, this probability is not a function of the different word senses. In our system, we set $\alpha_5 = 0.9$ because the hypernym relationship represents strong relationship between the senses and is similar to the hyponym relationship. Figure 4 shows an example of how the edge weights are computed.

We next consider the *meronym* (a.k.a. part-of) relationship in WordNet. For example, WordNet contains the information that the sense of the word "back": "a support that you can lean against ..." and the sense of the word "leg": "one of the supports for a piece of furniture" are both meronyms of the first sense of the word "chair". In other words, back and legs are building parts of a chair. This information can be represented in a phrase graph, as show in Figure 5. In general, the weight of an edge is set to α_6/n, where n is the number of meronyms. The parameter α_6 is used to denote the probability that a user that is interested in a sense of a word form is also interested in one of its *holonyms* (X is a holonym of Y exactly when Y is a meronym of X). In our system, $\alpha_6 = 0.5$ because the meronym relationship is not as strong as the hyponym relationship.

Similarly, we define α_7 to be equal to the probability that a user that is interested in a sense of a word form is also interested in one of its holonyms. We will draw an edge from every sense of a word form to all its holonyms, where the weight of each edge is set to α_7. In our system, $\alpha_7 = 0.5$ because the holonym relationship is similar to the meronym relationship. The edges to the node "a seat for one person ..." in Figure 5 show examples of how the weights for the holonym relationship are represented in the graph.

WordNet also defines two relationships for adjectives: *related to* and *similar to*. For example, the first sense of the adjective "slow" has definition: "not moving quickly...", while the first sense of the adjective "fast" has the definition: "acting or moving or capable of acting or moving quickly". WordNet specifies that the two senses are related to each other. We will draw an edge between the two senses with weight $\alpha_8 = 0.6$ - see Figure 6.

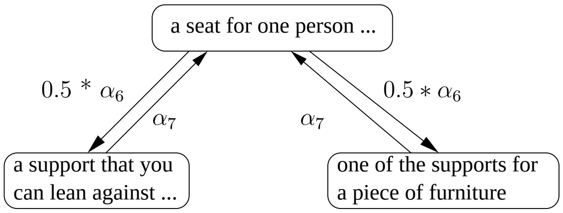
a seat for one person ...
0.5 * α_6
α_7
α_7
$0.5 * \alpha_6$
a support that you can lean against ...
one of the supports for a piece of furniture

Fig. 5. Representing Meronyms and Holonyms

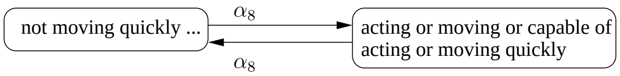

Fig. 6. Representing the *related to* Relationship Between Adjectives

Lastly, WordNet defines the *similar to* relations between adjectives. We draw edges with weight $\alpha_9 = 0.8$ between similar sense because the similar to relationship is stronger than the related to relationship. For example, WordNet contains the information that sense for the word "frequent": "coming at short intervals or habitually" and the sense for the word "prevailing": "most frequent or common" are similar to each other. We will therefore draw edges with weight α_9 between the two senses - see Figure 7.

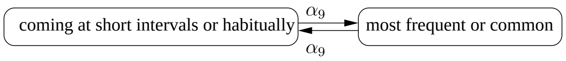

Fig. 7. Representing the *similar to* Relationship Between Adjectives

IV. Measuring Semantic Similarity

The phrase graph is used to represent the conditional probability that a user is interested in a phrase given that they are interested in a neighboring phrase. We compute the directional similarity between two nodes using the following formula.

$$A \rightarrow_s C = \sum_{Pt \text{ is a cycleless path from A to C}} P_{Pt}(C|A) \tag{1}$$

$$P_{Pt}(C|A) = \prod_{(n_1,n_2) \text{ is an edge in } Pt} P(n_2|n_1) \tag{2}$$

Informally, we compute the directional similarity between two nodes in the graph as the sum of all the paths between the two nodes, where we eliminate cycles from the paths. Each path provides evidence about the similarity between the two phrases. We compute the similarity between two nodes along a path as the product of the weights of the edges along the path, which follows the Markov chain model. Since the weight of an edge along the path is almost always smaller than one (i.e, equal to one only in rear circumstances), the value of the conditional probability will degrade as the length of the path increases. This is desirable behavior because a longer path provides less evidence about the similarity of the two end nodes.

Next, we present two functions for measuring similarity, where our online implementation uses the second function (see Equation 4). The linear function for computing similarity is shown in Equation 3.

$$|w_1, w_2|_{lin} = min(\alpha_{10}, \frac{w_1 \rightarrow_s w_2 + w_2 \rightarrow_s w_1}{2}) * \frac{1}{\alpha_{10}} \quad (3)$$

The minimum function was used to cap the value of the similarity function at 1. α_{10} is a coefficient that amplifies the available evidence. The next section shows how the value of α_{10} affects the correlation between the results of the system and that of human judgement.

The second similarity function is inverse logarithmic, that is, it amplifies the smaller values. It is shown in Equation 4. The *norm* function simply multiplies the result by a constant (i.e., $-ln(\alpha_{10})$) so that the resulting value is in the range [0,1]. Note that the *norm* function does not affect the correlation value.

$$|w_1, w_2|_{log} = norm(\frac{-1}{ln(min(\alpha_{10}, \frac{w_1 \rightarrow_s w_2 + w_2 \rightarrow_s w_1}{2}))}) \quad (4)$$

Given two nodes, the similarity between them is computed by performing a breadth-first traversal of the graph from each node in parallel. Common nodes between the two traversals identify paths between the two nodes. In our system, we set the depth of the search to paths of length 300; increasing this number had no meaningful effect on the similarity score.

V. Experimental Evaluation

The system consists of the two programs: one that creates the phrase graph and one that queries the phrase graph. Both programs were implemented in Java and can be found at http://softbase.ipfw.edu/SimilaritySystem. We used the Java API for WordNet Searching (JAWS) to connect to WordNet. The interface was developed by Brett Spell ([23]). All experiments were performed on a Silicon Graphics UV10 Linux machine. The Web interface of the system was created using JavaServer Pages (JSP)([24]). It takes about five minutes to build the phrase graph and save it to the hard disk and about one minute to load it from the hard disk. The average time for computing the similarity distance between two words is ten seconds.

We used the system to compute the similarity of 28 pairs of words from the Miller and Charles study. The study presented the words to humans and computed the mean score of the human ranking. The results are shown in Table I. The table shows the result for α_{10} equal to 0.1, 0.2, and 0.3. To save space, the actual results for the 28 pairs are posted on the website from the previous paragraph. As Table II suggests, the correlation drops as the value of α_{10} diverges from these values.

The values $\{\alpha_i\}_{i=1}^{9}$ seem to have little effect on the correlation. For example, setting all nine parameters to 1 worsened the correlation result by only two percent. The α_{10} coefficient has the most effect on the correlation result.

algorithm	*correlation*
M&C means	1.00
$\vert \cdot \vert_{lin}$, $\alpha_{10} = 0.1$	0.87
$\vert \cdot \vert_{lin}$, $\alpha_{10} = 0.2$	0.87
$\vert \cdot \vert_{lin}$, $\alpha_{10} = 0.3$	0.85
$\vert \cdot \vert_{log}$, $\alpha_{10} = 0.1$	0.89
$\vert \cdot \vert_{log}$, $\alpha_{10} = 0.2$	0.90
$\vert \cdot \vert_{log}$, $\alpha_{10} = 0.3$	0.88

TABLE I
Similarity Results

Table II shows the result of the correlation with different values for α_{10}. Table III show how our results compare with other proposals for extracting semantic similarity between word forms from WordNet. The results are for $\alpha_{10} = 0.1$. As the table suggests, both our algorithms produce better results (i.e., closer correlation with the results from the human judgement experiment in [16]) than existing algorithms.

α_{10}	$\vert \cdot \vert_{lin}$	$\vert \cdot \vert_{log}$
0.1	0.87	0.89
0.2	0.87	0.90
0.3	0.85	0.88
0.4	0.83	0.85
0.5	0.81	0.82
0.6	0.79	0.78
0.7	0.78	0.73
0.8	0.76	0.67
0.9	0.75	0.58
1.0	0.73	N/A

TABLE II
Correlation Results for different values of α_{10} on the [16] benchmark

We explored how the coefficient α_{10} affects the quality of the result. We showed that we get the highest correlation with the results from the Miller and Charles study ([16]) when α_{10} is equal to 0.1 and 0.2 and when we use the logarithmic similarity metric (respectively, 0.89 and 0.90 correlation score). In order to avoid overfitting, we decided to check if similar results hold for a different benchmark. In particular, we used the WordSimilarity-353 dataset ([4]). It contains 353 word pairs. Thirteen humans were used to rate the similarity between each pair of words and give a score between 1 and 10 (10 meaning that the words have the same meaning and 1 meaning that the words are unrelated). The average similarity rating for each word pair was recorded. Table IV shows the correlation between our linear and logarithmic algorithms with different

algorithm	*correlation*
Hirst and St-Onge ([7])	0.74
Leacock and Chodorow ([13])	0.82
Resnik ([19])	0.77
Jiang and Conrath ([10])	0.85
Lin ([14])	0.83
$\vert \cdot \vert_{lin}$	0.87
$\vert \cdot \vert_{log}$	0.89

TABLE III
Correlation Results with [16]

values of α_{10} and the results from the WordSimilarity-353 benchmark.

α_{10}	$\lvert\cdot\rvert_{lin}$	$\lvert\cdot\rvert_{log}$
0.1	0.46	0.49
0.2	0.41	0.46
0.3	0.38	0.43
0.4	0.37	0.40
0.5	0.36	0.39
0.6	0.35	0.37
0.7	0.35	0.34
0.8	0.34	0.31
0.9	0.34	0.27
1.0	0.33	N/A

TABLE IV
CORRELATION RESULTS FOR DIFFERENT VALUES OF α_{10} ON THE [4] BENCHMARK

Table V shows how our system compares with eight existing systems that have documented their performance on the WordSimilarity-353 benchmark. The results of our system are for $\alpha_{10} = 0.1$. As the table shows, our system produces better results for 7 out of the 8 systems and comparable results for system [1]. However, note that the algorithm from [1] uses information from the Web, while our algorithm only uses information from WordNet. As we extend our system to use information from Wikipedia, we hope to further improve the quality of the results of our system.

VI. CONCLUSION

We presented an algorithm for building a semantic corpus from WordNet. We verified the algorithm by showing that it can be used to compute the semantic similarity between phrases and we experimentally verified that the algorithm can produce better quality result than existing algorithms on the Charles and Miller and WordSimilarity-353 word pairs benchmarks. We believe that we outperform existing algorithms because our algorithm processes not only structured data, but also natural language. Our next area for future research is to extend the semantic corpus to incorporate data from Wikipedia.

algorithm	*correlation*
Jarmasz ([8])	0.27
Hirst and St-Onge ([7])	0.34
Jiang and Conrath ([10])	0.34
Strube and Ponzetto ([26])	0.19-0.48
Leacock and Chodrow ([13])	0.36
Lin ([14])	0.36
Resnik ([19]	0.37
Bollegala et al. ([1])	0.50
$\lvert\cdot\rvert_{lin}$	0.46
$\lvert\cdot\rvert_{log}$	0.49

TABLE V
CORRELATION RESULTS WITH [4]

REFERENCES

[1] D. Bollegala, Y. Matsuo, and M. Ishizuka. A Relational Model of Semantic Similarity Between Words Using Automatically Extracted Lexical Pattern Clusters from Web. *Conference on Empirical Methods in Natural Language Processing*, 2009.

[2] L. Burnard. Reference Guide for the British National Corpus (XML Edition). *http://www.natcorp.ox.ac.uk*, 2007.

[3] R. L. Cilibrasi and P. M. Vitanyi. The Google Similarity Distance. *IEEE ITSOC Informtaion Theory Workshop*, 2005.

[4] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131, January 2002.

[5] C. Fox. Lexical Analysis and Stoplists. *Information Retrieval: Data Structures and Algorithms*, pages 102–130, 1992.

[6] W. Frakes. Stemming Algorithms. *Information Retrieval: Data Structures and Algorithms*, pages 131–160, 1992.

[7] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *Fellbaum*, pages 305–332, 1998.

[8] M. Jarmasz. Roget's Thesaurus as a Lexical Resource for Natural Language Processing. *Master's thesis, University of Ottawa*, 1993.

[9] G. Jeh and J. Widom. SimRank: A Measure of Structural-context Similarity. *Proceedings of the Eight ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543, 2002.

[10] J. Jiang and D. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *Proceedings on International Conference on Research in Computational Linguistics*, pages 19–33, 1997.

[11] R. Knappe, H. Bulskov, and T. Andreasen. Similarity Graphs. *Fourteenth International Symposium on Foundations of Intelligent Systems*, 2003.

[12] S. Kulkami and D. Caragea. Computation of the Semantic Relatedness Between Words Using Concept Clouds. *International Conference of Knowledge Discovery and Information Retrieval*, 2009.

[13] C. Leacock and M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. *WordNet: An electronic lexical database*, pages 265–283, 1998.

[14] D. Lin. An Information-theoretic Definition of Similarity. *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, 1998.

[15] M.F.Porter. An Algorithm for Suffix Stripping. *Readings in Information Retrieval*, pages 313–316, 1997.

[16] G. Miller and W. Charles. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processing*, 6(1):1–28, 1991.

[17] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[18] R. Pan, Z. Ding, Y. Yu, and Y. Peng. A Bayesian Network Approach to Ontology Mapping. *Proceedings of the Fourth International Semantic Web Conference*, 2005.

[19] P.Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[20] R. Rada, H. Mili, E. Bickness, and M. Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.

[21] Q. Rajput and S. Haider. Use of Bayesian Networks in Information Extraction from Unstructured Data Sources. *Proceedings of International Conference on Ontological and Semantic Engineering*, pages 325–331, 2009.

[22] Simone Paolo Ponzetto and Michael Strube. Deriving a Large Scale Taxonomy from Wikipedia. *22nd International Conference on Artificial Intelligence*, 2007.

[23] B. Spell. Java API for WordNet Searching (JAWS). *http://lyle.smu.edu/ tspell/jaws/index.html*, 2009.

[24] L. Stanchev. Similarity Software. *http://softbase.ipfw.edu:8080/Similarity*, 2012.

[25] M. Steyvers and J. Tenenbaum. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, 29(1):41–78, 2005.

[26] M. Strube and S.P.Ponzetto. Wikirelate! Computing Semantic Relatedness using Wikipedia. *Association for the Advancement of Artificial Intelligence Conference*, 2006.

[27] Z. Wu and M. Palmer. Verb Semantics and Lexical selection. *Annual Meeting of the Association for Computational Linguistics*, pages 133–138, 1994.

[28] D. Yang and D. M. Powers. Measuring Semantic Similarity in the Taxonomy of WordNet. *Australian Computer Science Conference*, pages 315–322, 2005.