SEMILAR: The Semantic Similarity Toolkit

Vasile Rus
The University of Memphis
Department of Computer Science
Memphis, TN, 38152, USA
vrus@memphis.edu

Mihai Lintean
The University of Memphis
Department of Computer Science
Memphis, TN, 38152, USA
mclinten@memphis.edu

Rajendra Banjade
The University of Memphis
Department of Computer Science
Memphis, TN, 38152, USA
rbanjade@memphis.edu

Nobal Niraula
The University of Memphis
Department of Computer Science
Memphis, TN, 38152, USA
nbnraula@memphis.edu

Dan Stefanescu
The University of Memphis
Department of Computer Science
Memphis, TN, 38152, USA
dstfnscu@memphis.edu

Abstract

We present here SEMILAR, a SEMantic simILARity toolkit. SEMILAR implements a number of algorithms for assessing the semantic similarity between two texts. It is available as a Java library and as a Java standalone application offering GUI-based access to the implemented semantic similarity methods. Furthermore, it offers facilities for manual semantic similarity annotation by experts through its component SEMILAT (a SEMantic simILarity Annotation Tool).

1 Introduction

We present in this paper the design and implementation of SEMILAR, a SEMantic simILARity toolkit. SEMILAR includes implementations of a number of algorithms proposed over the last decade or so to address various instances of the general problem of text-to-text semantic similarity. Semantic similarity is the widely used approach to language understanding. It is a practical alternative to the true understanding approach, which is intractable as it requires world knowledge, a yet-to-be-solved problem in Artificial Intelligence.

As a concrete example of a semantic similarity task, we show below an instance from the Microsoft Research Paraphrase Corpus (Dolan, Quirk, and Brockett, 2004).

Text A: *York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.*
Text B: *York had no problem with MTA's saying the decision to shift funds was within its powers.*

Given such two texts, the paraphrase identification task is about automatically assessing whether Text A is a paraphrase of, i.e. has the same meaning as, Text B. The example above is a positive instance, meaning that Text A is a paraphrase of Text B and vice versa.

The importance of semantic similarity in Natural Language Processing (NLP) is highlighted by the diversity of datasets and shared task evaluation campaigns (STECs) that have been proposed over the last decade (Dolan, Quirk, and Brockett, 2004; McCarthy & McNamara, 2008; Agirre et al., 2012).

These datasets include instances from various applications. Indeed, there is a need to identify and quantify semantic relations between texts in many applications. For instance, paraphrase identification, an instance of the semantic similarity problem, is an important step in a number of applications including Natural Language Generation, Question Answering, and dialogue-based Intelligent Tutoring Systems. In Natural Language Generation, paraphrases are a method to increase diversity of generated text (Iordanskaja et al. 1991). In Question Answering, multiple answers that are paraphrases of each other could be considered as evidence for the correctness of the answer (Ibrahim et al. 2003). In Intelligent Tutoring Systems (Rus et al., 2009; Lintean et al., 2010; Lintean, 2011), paraphrase identification is useful to assess whether students' articulated answers to deep questions (e.g. conceptual physics questions) are similar-to/paraphrases-of ideal answers.

Generally, the problem of semantic similarity between two texts, denoted text A and text B, is defined as quantifying and identifying the presence of semantic relations between the two texts, e.g. to what extent text A has the same meaning as or is a paraphrase of text B (paraphrase relation; Dolan, Quirk, and Brockett, 2004). Other semantic relations that have been investigated systematically in the recent past are entailment, i.e. to what extent text A entails or logically infers text B (Dagan, Glickman, & Magnini, 2004), and elaboration, i.e. to what extent text B is an elaboration of text A (McCarthy & McNamara, 2008).

Semantic similarity can be broadly construed between texts of any size. Depending on the granularity of the texts, we can talk about the following fundamental text-to-text similarity problems: word-to-word similarity, phrase-to-phrase similarity, sentence-to-sentence similarity, paragraph-to-paragraph similarity, or document-to-document similarity. Mixed combinations are also possible such as assessing the similarity of a word to a sentence or a sentence to a paragraph. For instance, in summarization it might be useful to assess how well a sentence summarizes an entire paragraph.

2 Motivation

The problem of word-to-word similarity has been extensively studied over the past decades and a word-to-word similarity library (WordNet Similarity) has been developed by Pedersen and colleagues (Pedersen, Patwardhan, & Michelizzi, 2004).

Methods to assess the semantic similarity of larger texts, in particular sentences, have been proposed over the last decade (Corley and Mihalcea, 2005; Fernando & Stevenson, 2008; Rus, Lintean, Graesser, & McNamara 2009). Androutsopoulos & Malakasiotis (2010) compiled a survey of methods for paraphrasing and entailment semantic relation identification at sentence level. Despite all the proposed methods to assess semantic similarity between two texts, no semantic similarity library or toolkit, similar to the WordNet library for word-to-word similarity, exists for larger texts. Given the importance of semantic similarity, there is an acute need for such a library and toolkit. The developed SEMILAR library and toolkit presented here fulfills this need.

In particular, the development of the semantic similarity toolkit SEMILAR has been motivated by the need for an integrated environment that would provide:

- easy access to implementations of various semantic similarity approaches from the same user-friendly interface and/or library.
- easy access to semantic similarity methods that work at different levels of text granularity: word-to-word, sentence-to-sentence, paragraph-to-paragraph, document-to-document, or a combination (SEMILAR integrates word-to-word similarity measures).
- authoring methods for semantic similarity.
- a common environment for that allows systematic and fair comparison of semantic similarity methods.
- facilities to manually annotate texts with semantic similarity relations using a graphical user interface that make such annotations easier for experts (this component is called SEMILAT component - a SEMantic similarity Annotation Tool).

SEMILAR is thus a one-stop-shop for investigating, annotating, and authoring methods for the semantic similarity of texts of any level of granularity.

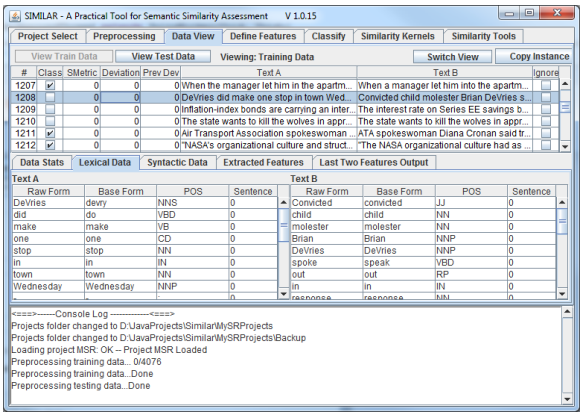

Figure 1. Snapshot of SEMILAR. The Data View tab is shown.

3 SEMILAR: The Semantic Similarity Toolkit

The authors of the SEMILAR toolkit (see Figure 1) have been involved in assessing the semantic similarity of texts for more than a decade. During this time, they have conducted a careful requirements analysis for an integrated software toolkit that would integrate various methods for semantic similarity assessment. The result of this effort is the prototype presented here. We briefly present the components of SEMILAR next and then describe in more detail the core component of SEMILAR, i.e. the set of semantic similarity methods that are currently available. It should be noted that we are continuously add new semantic similarity methods and features to SEMILAR.

The SEMILAR toolkit includes the following components: project management; data view-browsing-visualization; preprocessing (e.g., collocation identification, part-of-speech tagging, phrase or dependency parsing, etc.), semantic similarity methods (word-level and sentence-level), classification components for qualitative decision making with respect to textual semantic relations **(**naïve Bayes, Decision Trees, Support Vector Machines, and Neural Network), kernel-based methods **(**sequence kernels, word sequence kernels, and tree kernels; as of this writing, we are still implementing several other tree kernel methods); debugging and testing facilities for model selection; and annotation components (allows domain expert to manually annotate texts with semantic relations using GUI-based facilities; Rus et al., 2012). For space reasons, we only detail next the core component, i.e. the text-to-text similarity algorithms currently available in SEMILAR.

4 The Semantic Similarity Methods Available in SEMILAR

The core component of SEMILAR is a set of text-to-text semantic similarity methods. We have implemented methods that handle both unidirectional similarity measures as well as bidirectional similarity measures. For instance, the semantic relation of entailment between two texts is unidirectional (a text T logically entails a hypothesis text H but H does not entail T) while the paraphrase relation is bidirectional (text A has same meaning as text B and vice versa).

Lexical Overlap. Given two texts, the simplest method to assess their semantic similarity is to compute lexical overlap, i.e. how many words they have in common. There are many lexical overlap variations. Indeed, a closer look at lexical overlap reveals a number of parameters that turns the simple lexical overlap problem into a large space of possibilities. The parameters include preprocessing options (collocation detection, punctuation, stopword removal, etc.), filtering options (all words, content words, etc.), weighting schemes (global vs. local weighting, binary weighting, etc.), and normalization factors (largest text, weighted average, etc). A total of 3,456 variants of lexical overlap can be generated by different parameter settings in SEMILAR. Lintean (2011) has shown that performance on lexical overlap methods on the tasks of paraphrase identification and textual entailment tasks can vary significantly depending on the selected parameters. Some lexical overlap variations lead to performance results rivaling more sophisticated, state-of-the-art methods.

It should be noted that the overlap category of methods can be extended to include N-gram overlap methods (see the N-gram overlap methods proposed by the Machine Translation community such as BLEU and NIST). SEMILAR offers bigram and unigram overlap methods including the BLEU and NIST scores.

A natural approach to text-to-text similarity methods is to rely on word-to-word similarity measures. Many of the methods presented next compute the similarity of larger texts using individual word similarities.

Mihalcea, Corley, & Strappavara (2006; MCS) proposed a *greedy* method based on word-to-word similarity measures. For each word in text A (or B) the maximum similarity score to any word in the other text B (or A) is used. An idf-weighted average is then computed as shown in the equation below.

$$sim(T1,T2) = \frac{1}{2}\left(\frac{\sum_{w\in\{T_1\}} \max\{Sim(w,T_2)*idf(w)\}}{\sum_{w\in\{T_1\}} idf(w)} + \frac{\sum_{w\in\{T_2\}} \max\{Sim(w,T_1)*idf(w)\}}{\sum_{w\in\{T_2\}} idf(w)}\right)$$

The word-to-word similarity function *sim(w, T)* in the equation above can be instantiated to any word-to-word similarity measure (e.g. WordNet similarities or LSA). To be able to use the word-to-word measures that rely on WordNet, one must map words in the input texts onto concepts in WordNet, i.e. word sense disambiguation (WSD) is needed. As of this writing, SEMILAR addresses the issue in two simple ways: (1) selecting the most frequent sense for each word, which is sense #1 in WordNet, and (2) using all the senses for each word and then take the maximum (or average) of the relatedness scores for each pair of word senses. We label the former method as ONE (sense one), whereas the latter is labeled as ALL-MAX or ALL-AVG (all senses maximum score or all senses average score, respectively). Furthermore, most WordNet-based measures only work within a part-of-speech category, e.g. only among nouns or only among verbs.

Other types of word-to-word measures, such as those based on Latent Semantic Analysis or Latent Dirichlet Allocation, do not have a WSD challenge.

Rus and Lintean (2012; Rus-Lintean-Optimal Matching or ROM) proposed an *optimal* solution for text-to-text similarity based on word-to-word similarity measures. The optimal lexical matching is based on the optimal assignment problem, a fundamental combinatorial optimization problem which consists of finding a maximum weight matching in a weighted bipartite graph.

Given a weighted complete bipartite graph $G = X \cup Y; X \times Y)$, where edge xy has weight $w(xy)$, the optimal assignment problem is to find a matching M from X to Y with maximum weight.

A typical application is about assigning a group of workers, e.g. words in text A in our case, to a set of jobs (words in text B in our case) based on the expertise level, measured by $w(xy)$, of each worker at each job. By adding dummy workers or jobs we may assume that X and Y have the same size, *n*, and can be viewed as $X = \{x_1, x_2, \dots, x_n\}$ and Y = $\{y_1, y_2, \dots, y_n\}$. In the semantic similarity case, the weight $w(xy)$ is the word-to-word similarity between a word x in text A and a word y in text B.

The assignment problem can also be stated as finding a permutation π of {1, 2, 3, … , n} for which $\sum_{i=1}^{n} w(x_i y_{\pi(i)})$ is maximum. Such an assignment is called optimum assignment. An algorithm, the Kuhn-Munkres method (Kuhn, 1955), has been proposed that can find a solution to the optimum assignment problem in polynomial time.

Rus and colleagues (Rus et al., 2009; Rus & Graesser, 2006; Rus-Syntax-Negation or RSN) used a lexical overlap component combined with syntactic overlap and negation handling to compute an unidirectional subsumption score between two sentences, T (Text) and H (Hypothesis), in entailment recognition and student input assessment in Intelligent Tutoring Systems. Each text is regarded as a graph with words as nodes/vertices and syntactic dependencies as edges. The subsumption score reflects how much a text is subsumed or contained by another. The equation below provides the overall subsumption score, which can be averaged both ways to compute a similarity score, as opposed to just the subsumption score, between the two texts.

$$subsump(T,H) = \left(\alpha \times \frac{\sum_{V_h\in H_v} \max_{V_t\in T_v} match(V_h,V_t)}{|V_h|} + \beta \times \frac{\sum_{E_h\in H_e} \max_{E_t\in T_e} match(E_h,E_t)}{|E_h|}\right) \times \frac{(1+(-1)^{\#neg\_rel})}{2}$$

The lexical component can be used by itself (given a weight of 1 with the syntactic component given a weight of 0) in which case the similarity between the two texts is just an compositional extension of word-to-word similarity measures. The *match* function in the equation can be any word-to-word similarity measure including simple word match, WordNet similarity measures, LSA, or LDA-based similarity measures.

Fernando and Stevenson (FST; 2008) proposed a method in which similarities among all pairs of words are taken into account for computing the similarity of two texts. Each text is represented as a binary vector (1 – the word occurs in the text;

0 – the word does not occur in the text). They use a similarity matrix operator W that contains word-to-word similarities between any two words.

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} W \vec{b}^T}{|\vec{a}||\vec{b}|}$$

Each element w_{ij} represents the word-level semantic similarity between word a_i in text A and word b_j in text B. Any word-to-word semantic similarity measure can be used.

Lintean and Rus (2010; weighted-LSA or *w*LSA) extensively studied methods for semantic similarity based on Latent Semantic Analysis (LSA; Landauer et al., 2006). LSA represents words as vectors in a 300-500 dimensional LSA space. An LSA vector for larger texts can be derived by vector algebra, e.g. by summing up the individual words' vectors. The similarity of two texts A and B can be computed using the cosine (normalized dot product) of their LSA vectors. Alternatively, the individual word vectors can be combined through weighted sums. Lintean and Rus (2010) experimented with a combination of 3 local weights and 3 global weights. All these versions of LSA-based text-to-text similarity measures are available in SEMILAR.

SEMILAR also includes a set of similarity measures based on the unsupervised method **Latent Dirichlet Allocation (LDA; Blei, Ng, & Jordnan, 2003; Rus, Banjade, & Niraula, 2013).** LDA is a probabilistic generative model in which documents are viewed as distributions over a set of topics (θ_d - text *d*'s distribution over topics) and topics are distributions over words (φ_t – topic *t*'s distribution over words). That is, each word in a document is generated from a distribution over words that is specific to each topic.

A first LDA-based semantic similarity measure among words would then be defined as a dot-product between the corresponding vectors representing the contributions of each word to a topic ($\varphi_t(w)$ – represents the probability of word *w* in topic *t*). It should be noted that the contributions of each word to the topics does not constitute a distribution, i.e. the sum of contributions is not 1. Assuming the number of topics T, then a simple word-to-word measure is defined by the formula below.

$$LDA-w2w(w,v) = \sum_{t=1}^{T} \varphi_t(w)\varphi_t(v)$$

More global text-to-text similarity measures could be defined in several ways as detailed next.

Because in LDA a document is a distribution over topics, the similarity of two texts needs to be computed in terms of similarity of distributions. The Kullback-Leibler (KL) divergence defines a distance, or how dissimilar, two distributions *p* and *q* are as in the formula below.

$$HD(p,q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{1}^{T} (\sqrt{p_i} - \sqrt{q_i})^2}$$

If we replace *p* with θ_d (text/document *d*'s distribution over topics) and *q* with θ_c (text/document *c*'s distribution over topics) we obtain the KL distance between two documents (documents *d* and *c* in our example). The KL distance has two major problems. In case q_i is zero KL is not defined. Then, KL is not symmetric. The Information Radius measure (IR) solves these problems by considering the average of p_i and q_i as below. Furthermore, the IR can be transformed into a symmetric similarity measure as in the following (Dagan, Lee, & Pereira, 1997):

$$SIM(p,q) = 10^{-\delta IR(c,d)}$$

The Hellinger and Manhattan distances between two distributions are two other options that avoid the shortcomings of the KL distance. Both are options are implemented in SEMILAR.

LDA similarity measures between two documents or texts *c* and *d* can also include similarity of topics. That is, the text-to-text similarity is obtained multiplying the similarities between the distribution over topics (θ_d and θ_c) and distribution over words (φ_{t1} and φ_{t2}). The similarity of topics can be computed using the same methods illustrated above as the topics are distributions over words (for all the details see Rus, Banjade, & Niraula, 2013).

The last method we present is a semantic similarity method based on **the Quadratic Assignment Problem (QAP).** The QAP method aims at finding an optimal assignment from words in text A to words in text B, based on individual word-to-word similarity measures, while simultaneously maximizing the match between the syntactic dependencies of the words.

The Koopmans-Beckmann (1957) formulation of the QAP problem best fits this purpose. The goal of the original QAP formulation, in the domain of economic activity, was to minimize the objective function QAP shown below where matrix F de-

scribes the flow between any two facilities, matrix D indicates the distances between locations, and matrix B provides the cost of locating facilities to specific locations. F, D, and B are symmetric and non-negative.

$$\min QAP(F, D, B) = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{i,j} d_{\pi(i)\pi(j)} + \sum_{i=1}^{n} b_{i,\pi(i)}$$

The $f_{i,j}$ term denotes the flow between facilities *i* and *j* which are placed at locations $\pi(i)$ and $\pi(j)$, respectively. The distance between these locations is $d_{\pi(i)\pi(j)}$. In our case, F and D describe dependencies between words in one sentence while B captures the word-to-word similarity between words in opposite sentences. Also, we have weighted each term in the above formulation and instead of minimizing the sum we are maximizing it resulting in the formulation below.

$$\max QAP(F, D, B) = \alpha \sum_{i=1}^{n}\sum_{j=1}^{n} f_{i,j} d_{\pi(i)\pi(j)} + (1-\alpha) \sum_{i=1}^{n} b_{i,\pi(i)}$$

5 Discussion and Conclusions

The above methods were experimented with on various datasets for paraphrase, entailment, and elaboration. For paraphrase identification, the QAP method provides best accuracy results (=77.6%) on the MSRP corpus.

References

Androutsopoulos, I. & Malakasiotis, P. 2010. A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research, 38:135-187.

Agirre, E., Cer, D., Diab, M., & Gonzalez-Agirre, A. (2012). SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity, First Joint Conference on Lexical and Computational Semantics (*SEM), Montreal, Canada, June 7-8, 2012.

Blei, D.M., Ng, A.Y., & Jordan, M.I. 2003. Latent dirichlet allocation, The Journal of Machine Learning Research 3, 993-1022.

Lintean, M., Moldovan, C., Rus, V., & McNamara D. (2010). The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis. Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference. Daytona Beach, FL.

Lintean, M. (2011). Measuring Semantic Similarity: Representations and Methods, PhD Thesis, Department of Computer Science, The University of Memphis, 2011.

Corley, C., & Mihalcea, R. (2005). Measuring the Semantic Similarity of Texts. In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. Ann Arbor, MI.

Dagan, I., Glickman, O., & Magnini, B. (2004). The PASCAL Recognising textual entailment Challenge. In Quinorero-Candela, J.; Dagan, I.; Magnini, B.; d'Alche-Buc, F. (Eds.), *Machine Learning Challenges.* Lecture Notes in Computer Science, Vol. 3944, pp. 177-190, Springer, 2006.

Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING*, Geneva, Switzerland.

Fernando, S. & Stevenson, M. (2008). A semantic similarity approach to paraphrase detection, *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium*.

Ibrahim, A., Katz, B., & Lin, J. (2003). Extracting structural paraphrases from aligned monolingual corpora In Proceedings of the Second International Workshop on Paraphrasing, (ACL 2003).

Iordanskaja, L., Kittredge, R., & Polgere, A. (1991). Natural Language Generation in Artificial Intelligence and Computational Linguistics. *Lexical selection and paraphrase in a meaning-text generation model*, Kluwer Academic.

McCarthy, P.M. & McNamara, D.S. (2008). User-Language Paraphrase Corpus Challenge https://umdrive.memphis.edu/pmmccrth/public/ParaphraseCorpus/Paraphrase site.htm. Retrieved 2/20/2010 online, 2009.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity - Measuring the Relatedness of Concepts, In the Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), pp. 1024-1025, July 25-29, 2004, San Jose, CA (Intelligent Systems Demonstration).

Rus, V., Lintean M., Graesser, A.C., & McNamara, D.S. (2009). Assessing Student Paraphrases Using Lexical Semantics and Word Weighting. In Proceedings of the 14th International Conference on Artificial Intelligence in Education, Brighton, UK.

Rus, V., Lintean, M., Moldovan, C., Baggett, W., Niraula, N., Morgan, B. (2012). The SIMILAR Corpus: A Resource to Foster the Qualitative Understanding of Semantic Similarity of Texts, In Semantic Relations II: Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012), May 23-25, Instanbul, Turkey.

Rus, V., Banjade, R., & Niraula, N. (2013). Similarity Measures based on Latent Dirichlet Allocation, The 14th International Conference on Intelligent Text Procesing and Computational Linguistics, March 24-30, 2013, Samos, Greece.