Robust, Light-weight Approaches to compute Lexical Similarity

Quang Do, Dan Roth, Mark Sammons, Yuancheng Tu, V.G.Vinod Vydiswaran
University of Illinois at Urbana-Champaign
Urbana, IL
{quangdo2, danr, mssammon, ytu, vgvinodv}@illinois.edu

Abstract

Most text processing systems need to compare lexical units – words, entities, semantic concepts – with each other as a basic processing step within large and complex systems. A significant amount of research has taken place in formulating and evaluating multiple similarity metrics, primarily between words. Often, such techniques are resource-intensive or are applicable only to specific use cases. In this technical report, we summarize some of our research work in finding robust, light-weight approaches to compute similarity between two spans of text. We describe two new measures to compute similarity, `WNSim` for word similarity, and `NESim` for named entity similarity, which in our experience have been more useful than more standard similarity metrics. We also present a technique, `Lexical Level Matching (LLM)`, to combine such token-level similarity measures to compute phrase- and sentence-level similarity scores. We have found `LLM` to be useful in a number of NLP applications; it is easy to compute, and surprisingly robust to noise.

1 Introduction

Many text understanding applications need to compute similarity between short documents or text snippets as one of their fundamental operations. For instance, in document summarization tasks, there is a need to identify and remove duplicate text snippets so as to maximize the diversity of information in a summary. Similarly, computing semantic similarity between snippets is essential in evaluating automated paraphrasing techniques (Kauchak and Barzilay, 2006; Connor and Roth, 2007). In textual entailment tasks, there is a need to decide if two sentences refer to the same concepts, even though they may be using different words to express them. Large news aggregators need to identify similar news reports so that they can cluster or remove duplicate news articles received from various news agencies.

In order to improve the accuracy of computing the similarity of short text segments, we must consider semantic relations between words so that we do not solely rely on exact matching of words. Indeed, words are not mutually independent, but to human readers have certain similarity with other words: very high for synonyms and other replaceable words, and quite low for unrelated words. A number of researchers have tried to design a similarity metric that captures this human intuition (Pedersen et al., 2004). However, words themselves are not the only basic units in language. A sequence of words may form named entities, phrasal verbs, or multi-word expressions, which have a distinct meaning of their own beyond the meanings of their component words. These units require their own similarity metrics, comparable to the word-similarity metrics, and convey significant information that must be used when finding similar documents or computing similarity between a pair of documents.

In this technical report, we propose two new metrics to compute similarity between sen-

tence constituents. The first metric, `WNSim`, is a word and multi-word expression similarity metric based on the WordNet hierarchy. It uses the links between words and synsets in WordNet across part-of-speech hierarchies to compute the similarity measure. The second metric, `NESim`, is a rule-based similarity metric between named entities that can optionally consider the type of named entities when computing the similarity score. In section 2.3, we present a simple yet robust technique called `Lexical Level Matching (LLM)` to combine such token-level similarity measures to compute sentence-level similarity. We claim that `LLM` is a robust measure of lexical similarity and can act as a baseline in many NLP tasks such as Textual Entailment and Paraphrasing. We present empirical evidence to support this claim in Sec. 3.

2 Proposed Similarity Measures

Semantic similarity between words has been studied extensively. One of the most influential works has been in building WordNet (Fellbaum, 1998). WordNet organizes words into synsets that are further linked to other synsets using hypernymy, meronymy, and other linguistic relations. There have been many similarity metrics proposed using the hierarchical structure of WordNet, especially for nouns (Pedersen et al., 2004).

2.1 `WNSim` measure for words and multi-words

We formulate a similarity measure, `WNSim`, over the WordNet hierarchy to compute similarity between words. For two words w_1 and w_2 in the WordNet hierarchy, `WNSim` finds the closest common ancestor of the words, sometimes referred to as *least common subsumer* (*lcs*). The similarity is then defined based on the distance of the words from the *lcs*, as follows:

$$\texttt{WNSim}(w_1, w_2) = \begin{cases} \theta^{\ell_1+\ell_2} & \text{if } \ell_1+\ell_2 \le k \\ \theta^{k} & \text{if } \ell_1+\ell_2 \le \alpha \times \text{depth of } lcs(w_1,w_2) \\ 0 & \text{otherwise} \end{cases}$$

This measure captures the key concepts of hierarchical similarity used in other WordNet-based similarity measures. It has 3 parameters: θ, k, and α. In the experiments, we empirically set them as $\theta = 0.3$, $k = 3$, and $\alpha = 0.667$, after manually searching over various values for these parameters.

The words are first converted to the same part-of-speech, by finding the base verb or noun form of the word, if available, before the appropriate WordNet hierarchy is considered. To compute the least common subsumer, we consider the synonymy-antonymy, hypernymy-hyponymy, and meronymy relations. If the path from the *lcs* to one of the words contains an antonymy relation, we reduce the similarity value by half and negate the score. Hence, under this scheme, synonyms get a score of 1.0 and antonyms get a similarity value of -0.5. Further, we compare the determiners and prepositions separately – if two words are determiners or prepositions, they get a similarity score of 0.5. Hence, this similarity measure gives a score in $[-1, 1]$ range. (The motivation here is to discount differences between words that tend to have little influence on overall similarity judgements – different prepositions, for example, may take on similar meanings based on context).

2.2 `NESim` measure for named entities

As with words, named entities need to be compared in a variety of NLP tasks, such as entity/schema matching and named co-reference discovery. For example, in the schema matching task, it is important to know that *George Bush* is the same as *Bush, George* or that *Mr. Smith* is different from *Mrs. Smith*. Several named entity metrics were developed incorporating inputs from statistical methods, databases, or artificial intelligence (Cohen et al., 2003). However, most of the existing approaches are limited in two aspects: (1) they do not take advantage of the named entity types when computing the similarity, and (2) they do not consider the semantics of the tokens in named entities.

We have found that the types and the semantics of the tokens in named entities play important roles in named entity metrics. For example, *George Washington* and *Washington* are similar if we know that they are two person names, but different if they are two locations, and more obviously, they are different if one refers to a location and the other refers to a person. Similarly, it would be simple to com-

String 1	Type	String 2	Type	JRWK	NESim
Joan Smith	PER	John Smith	PER	0.946	0.0
George Bush	-	Bush, George	-	0.0	1.0
Shiite	ORG	Shi'ite	ORG	0.922	0.922
United States	LOC	US	LOC	0.746	1.0
Paris	-	France	-	0.411	0.411
Wilbur T. Gobsmack	PER	Mr. Gobsmack	PER	0.78	0.95
National Science Foundation	ORG	NSF	ORG	0.62	1.0

Table 1: Similarity scores for different input combinations, using Jaro-Winkler (JRWK) and `NESim` measures. `NESim` also works when the entity type is absent, as shown by examples with entity type marked as "-".

pare these two names if we know that *George* is the first name and *Washington* is the last name. To address these two limitations, we incorporate the two main improvements specified below.

1**. Leveraging the types of named entities** in measuring their similarity. Named entity types are given by many named entity recognition packages. Standard types include *person*, *location*, and *organization*. If two names have different types, they should not be considered as similar. Therefore, our similarity computation depends on the named entity types. If two names are labeled as persons, they will be compared based on their identified first names, last names, and if available, their middle names. We also consider replacing nicknames with their original names in order to improve the coverage of our metric (e.g. Bob and Robert). Honorifics are also separated and identified, so that gender based comparison can be made. If two names are locations, the metric considers several standard ways of expressing locations, such as using abbreviations (e.g. `IL` for Illinois, or `VN` and `VNM` for Vietnam) and using a country-language look-up table (e.g. `Russia` and `Russian`.) For organizations, our metric is able to capture acronyms which are often used when an organization name is mentioned many times (e.g. `NATO` for North Atlantic Treaty Organization). If the type is not known, the metric tries the three types one-by-one, and returns the highest score.

2**. Parsing the input names** to identify the semantics of each token in the names. This is required especially for person names. In order to compare first names, last names, and middle names of persons, the metric parses the input names into fields and compares them separately. Person names are parsed using common cues in name format, such as names with or without commas (for first and last names), names with or without abbreviation (for first and middle names), etc. The metric's parser also identifies organization acronyms by combining the initial letters of name tokens. It is worth noting that there are several cases where this heuristic is not sufficient to form an acronym (e.g. `AIRTC` stands for Air Training Corps). The parser deals with this phenomenon by simply trying several combinations of the name tokens' first letters.

As a final back-off step, our metric uses edit distance metrics (viz. the one proposed by (Cohen et al., 2003)) to measure the similarity between named entities if none of the above conditions is matched.

Table 1 shows some similarity scores of the input names given by a metric using Jaro-Winkler distance (JRWK) ((Cohen et al., 2003)) and our method, `NESim`.

2.3 Computing sentence similarity using term-similarity metrics

The similarity between two sentences is computed based on the individual term-similarity as follows: First, both sentences are tokenized to find all semantic units, viz. named entities, phrasal verbs, multi-word expressions, and words. Then, the similarity metrics are applied based on the type of semantic units to match the units from one sentence to the most similar unit from the other sentence. At the end of this step, all semantic units map to their best counterparts from the other sentence. Finally, the sentence-level similarity

Particulars	MSR PP	RTE3
Number of sentence pairs	5801	1600
Number of positive pairs	3900 (67.2%)	822 (51.4%)
Number of negative pairs	1901 (32.8%)	778 (48.6%)
Training set	4077 (70.3%)	800 (50.0%)
Test set	1724 (29.7%)	800 (50.0%)
Number of unique words	11373	7248
Average length of sentences (in words)	13	14

Table 2: Dataset Characteristics

score is computed as the sum of the similarity scores of the matching pairs, normalized by the number of units matched. We refer to this measure as the `Lexical Level Matching (LLM)` score. For two sentences s_1 and s_2, such that $|s_1| \geq |s_2|$,

$$\texttt{LLM}(s_1, s_2) = \frac{\sum_{v \in s_2} \max_{u \in s_1} sim(u, v)}{|s_2|}$$

where $sim(u, v)$ is one of the similarity metrics defined over semantic units u and v.

3 Evaluation

To evaluate the proposed metrics, we compute `LLM` scores in three settings: using exact word matching (`LLM Exact`); using `WNSim` and named entity similarity together (`LLM WNSim`); and using `WUP` and named entity similarity metric together (`LLM WUP`). `WUP` is a hierarchical metric defined by (Wu and Palmer, 1994) and is made available as a word similarity metric over the WordNet hierarchy in WordNet::Similarity package (Pedersen et al., 2004).

3.1 Baseline measures for snippet similarity

To compare our approach with other similarity metrics defined over text snippets, we follow the analysis in (Achananuparp et al., 2008). We choose TF-IDF and Word Ordering measures as our baseline metrics for the analysis.

TF-IDF Instead of assigning a uniform weight, each word is assigned a weight equal to its inverse document frequency (*idf*). Such a measure gives higher importance to rare content words than to frequent function words (stopwords). The IDF for a term t depends inversely on the number of documents in which t occurs ($df(t)$) in a corpus C of size $N = |C|$. In our experiments, we use the standard IDF formulation:

$$idf_C(t) = \log \frac{N + 1}{df(t)}$$

The TF-IDF similarity of two snippets is computed as the product of term frequency (tf) and its *idf*, summed over all words common to the two snippets, and normalized by the norm of idf weights for individual snippets.

Word Ordering This measure gives importance to the order in which words appear in a snippet. The order information may be important, especially in paraphrasing and textual entailment tasks, since the subject and agent of an action need to be the same in both sentences. The *WordOrder* similarity measure is computed as

$$order(s_1, s_2) = 1 - \frac{||r_1 - r_2||}{||r_1 + r_2||}$$

where r_1, r_2 are word order vectors: each word in vector r_i has a weight corresponding to its position in the sentence s_i.

We evaluate the measures for two tasks. The first is to recognize paraphrases, and we show the results over the the MSR Paraphrase Corpus (Dolan et al., 2004). Secondly, we address the task of recognizing textual entailment using only lexical resources and show the results over PASCAL RTE3 Corpus (Dagan et al., 2006). These corpora have been used previously to evaluate sentence semantic measures (Achananuparp et al., 2008). The characteristics of the datasets used in the evaluation is given in Table 2.

3.2 Detecting Paraphrases

We compute the snippet similarity for all 5801 pairs of the MSR Paraphrase corpus, using

Metric	Accu	Prec	Rec	F_1
TF-IDF	0.686	0.696	0.948	0.803
WordOrder	0.705	0.730	0.891	0.802
LLM Exact	0.710	0.723	0.923	0.811
LLM WNSim	**0.711**	0.748	0.861	0.800
LLM WUP	0.708	0.729	0.897	0.805

Table 3: Performance of the metrics in detecting paraphrases

Metric	Accu	Prec	Rec	F_1
TF-IDF	0.568	0.571	0.645	0.605
WordOrder	0.535	0.536	0.713	0.612
LLM Exact	0.645	0.620	0.797	0.698
LLM WNSim	**0.651**	0.619	0.833	0.711
LLM WUP	0.644	0.634	0.725	0.697

Table 4: Performance of the metrics in recognizing textual entailment over RTE3 dataset.

the different measures defined in Sec. 2 and Sec. 3.1. After finding the similarity scores using the similarity metrics, we rank the documents on the similarity score and choose a threshold that maximizes the accuracy over the training data. We report the accuracy, precision, recall, and F_1 scores over the complete dataset. Accuracy measures the fraction of all instances that were labeled correctly, including both positive and negative instances. Precision measures the fraction of positively labeled instances that were correctly labeled, and Recall measures the fraction of positive instances that were correctly labeled. F_1 is the harmonic mean of the precision and recall values.

In this evaluation, Accuracy is the more appropriate measure, since it is important to recognize both positive and negative instances correctly.

The performance of the similarity metrics in classifying a sentence pair as paraphrases is summarized in Table 3. We see that `LLM WNSim` is the best measure among the semantic metrics. For this dataset, however, `LLM Exact` gave the best F_1 score.

3.3 Recognizing textual entailment

We compute the similarity metric over all 1600 pairs of the RTE3 corpus and follow similar evaluation strategy to the one described in Sec. 3.2. The results are summarized in Table 4. We see that `LLM WNSim` outperforms all other semantic metrics.

It must be pointed out that the evaluation uses only the semantic similarity score to classify a pair as being entailed or not. Many researchers have shown better performance scores for the task using additional knowledge sources. Our comparison is primarily to validate the understanding that snippet similarity is an important component in this task and, by itself, can perform fairly well (Roth and Sammons, 2007).

4 Conclusion

In this report, we propose lexical and rule-based measures, `WNSim` and `NESim`, to compute similarity between words and named entities, respectively. We demonstrate a robust technique, called `Lexical Level Matching (LLM)`, to combine similarity scores between lexical tokens to compute a sentence-level similarity score. We demonstrate the use of `LLM` to get competitive performance in recognizing textual entailment, and propose this as a robust baseline for the task.

References

Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. 2008. The Evaluation of Sentence Similarity Measures. In *DaWaK '08: Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, pages 305–316.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. In *IJCAI Workshop on Information Integration on the Web*.

Michael Connor and Dan Roth. 2007. Context Sensitive Paraphrasing with a Global Unsupervised Classifier. In *ECML*, pages 104–115.

I. Dagan, O. Glickman, and B. Magnini, editors. 2006. *The PASCAL Recognising Textual Entailment Challenge.*, volume 3944. Springer-Verlag, Berlin.

William Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING*.

C. Fellbaum. 1998. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of HLT-NAACL*, pages 455–462.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Demonstration Papers*, pages 38–41.

D. Roth and M. Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112, Prague, June. Association for Computational Linguistics.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexicon selection. In *ACL*, pages 133–138.