ATA-Sem: Chunk-based Determination of Semantic Text Similarity

Demetrios Glinos
Advanced Text Analytics, LLC
Orlando, Florida, USA
demetrios.glinos@advancedtextanalytics.com

Abstract

This paper describes investigations into using syntactic chunk information as the basis for determining the similarity of candidate texts at the semantic level. Two approaches were considered. The first was a corpus-based method that extracted lexical and semantic features from pairs of chunks from each sentence that were associated through a chunk alignment algorithm. The features were used as input to a classifier trained on the same features extracted from a corpus of gold standard training data. The second approach involved breadth-first chunk association and the application of a rule-based scoring algorithm. Both approaches were evaluated against the test data for the SemEval 2012 Semantic Text Similarity task. The results show that the rule-based chunk approach is superior.

1 Introduction

The task of determining whether two texts are similar in some sense has important applications in the field of natural language processing, including but not limited to document summarization (Evans, et al., 2005), plagiarism detection (Barrón-Cedeño, et al., 2009) and large corpus document retrieval (Charikar, 2002).

While textual similarity can be performed at the purely surface lexical level, as in the "simhash" clustering method described in (Moulton, 2010), similarity also applies at the semantic level, where conceptually similar texts may nevertheless be entirely dissimilar at the surface lexical level. For example, the phrases "restrict or confine" and "place limits on (extent or access)" share no words or morphological roots, yet mean very nearly the same thing at the semantic level.

The Semantic Textual Similarity (STS) task (Task #6) at SemEval-2012 (Agirre, et al., 2012) provided a forum for exploring these issues by furnishing training and evaluation data, and also a common standard for describing degrees of similarity, shown in Table 1.

Score	Description
5	The two sentences are completely equivalent, as they mean the same thing.
4	The two sentences are mostly equivalent, but some unimportant details differ.
3	The two sentences are roughly equivalent, but some important information differs/missing.
2	The two sentences are not equivalent, but share some details.
1	The two sentences are not equivalent, but are on the same topic.
0	The two sentences are on different topics.

Table 1. STS similarity scoring standard.

Our corpus-based chunk similarity method participated in the formal STS evaluation. Our rule-based method was completed after the submittal date, but we report on it here because the method does not involve training on a corpus, nor any parameter tuning, and because it significantly outperformed the corpus-based method.

The remainder of this paper is organized as follows. In the next section, we describe the common processing components for both methods. Section 3 then presents the corpus-based chunk method, followed in Section 4 by a discussion of the rule-based chunk similarity method. Section 5 concludes with a presentation of how the two methods performed against the STS test set, and offers some observations on the viability of chunk-based similarity determination.

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 547–551,
Montréal, Canada, June 7-8, 2012.

2 Common Processing Components

A common processing core supports both of the methods, comprising preprocessing components and also shared components for determining chunk-level similarity. The preprocessing components make use of the U.S. National Library of Medicine's Lexical Tools (NLM 2012) to perform ASCII conversion and tokenization. Candidate sentence pairs are then tagged and chunked using our own tagger and separate chunker, which were both trained on CONLL 2000 data using the CRF++ conditional random field toolkit (Taku-ku 2012). We use chunk labels that augment the standard BIO tags with appropriate Penn-Treebank phrasal tags, for example, "B-NP" and "B-ADVP".

Once the candidate sentences are chunked, the two methods diverge in their approach to classification. However, both approaches use the NLM Lexical Tools and WordNet (Fellbaum, 1998) for term expansion. The NLM's normalization tool is used to reduce terms to lower case, strip them of punctuation, stop words, diacritical marks, etc., and to expand the terms with lexical variants. WordNet's synonyms and hypernyms for the remaining terms are then added to expand the term lists for chunk-level comparisons.

3 Corpus-based Chunk Method

The corpus-based method employs a "chunk alignment" algorithm for selecting pairs of chunks for detailed comparison, one from each candidate sentence. The algorithm operates by initializing pointers to the first chunk in each sentence. Then, noting the chunk type for the indexed chunk in the shorter sentence, the algorithm marches down the longer sentence searching for the first chunk of the same type. Once it is found, the two chunks are marked for comparison and the index into the shorter sentence is incremented to the next chunk. The process repeats until no more chunk pairs can be associated. Figure 1 shows an example of chunk alignment.

The method generates the set of features shown in Table 2 based on the chunk-level comparisons. Features 2 to 4 contain numerical values representing the sums of "matching scores" from the aligned chunks. A four-valued matching score is assigned for each chunk comparison depending on the degree of chunk-level similarity. A value of "3" represents an exact term match or a match on a synonym. The value "2" is given if the head term of one of the chunks is in the hypernym tree for the other chunk. And a value of "1" is given if the two chunk heads have a common hypernym ancestor. The default value "0" is given if none of the above conditions is found. The numbers in brackets in the table identify the unigram features that are associated to compose trigram and 4-gram features, respectively.

Unigrams	0	Total # chunks in Sentence A
	1	Total # chunks in Sentence B
	2	Sum of aligned VP matching scores
	3	Sum of aligned NP matching scores
	4	Sum of aligned PP matching scores
	5	Number of VP chunks in A
	6	Number of NP chunks in A
	7	Number of PP chunks in A
	8	Number of VP chunks in B
	9	Number of NP chunks in B
	10	Number of PP chunks in B
Trigrams		[2, 5, 8], [3, 6, 9], [4, 7, 10]
4-grams		[0, 5, 6, 7], [1, 8, 9, 10]

Table 2. Similarity classifier features.

Once the feature vector is, it is passed to the text similarity classifier, which generates the 0-5 similarity score. The classifier was trained on the gold

Micron's numbers **NP**	also **ADVP**	marked **VP**	the first quarterly profit **NP**	in three years **PP**	for the DRAM manufacturer **PP**
Micron **NP**		has declared **VP**	its first quarterly profit **NP**	in three years **PP**	

Figure 1. Chunk Alignment Example

standard training data using the CRF++ toolkit using the same feature set described above.

4 Rule-based Chunk Method

The rule-based chunk similarity method employs a breadth-first search method for selecting candidate chunks for further comparison. The algorithm operates by selecting the first chunk of the sentence with the larger number of chunks. It then marches down each chunk of the shorter sentence looking for an exact term match or head term synonym match. If a match is found, a chunk-level score value of 3 is assigned, and the next chunk in the longer sentence is considered. If a match is not found, then a new search is performed, this time searching for a hypernym match. If a match is found in this second pass, a chunk score of 2 is assigned, and the next index chunk is considered. If not, then a third and final pass is performed searching for a related term match. If a match is found after this third pass, a chunk score of 1 is assigned; otherwise, the chunks are deemed dissimilar and receive a chunk score of zero.

We describe this algorithm as "breadth-first" because it has the effect of conducting up to three passes across all of the chunks of the target (shorter) sentence, looking for successively "looser" matches. For these purposes, we consider a hypernym match to be looser than an exact or synynym match, and a common-ancestor (related) term match to be looser than a hypernym match.

The chunk-level matching scores are accumulated in the above manner, just as for the corpus-based method. However, in this case, the results are used directly by the rule-based scoring algorithm. The scoring algorithm treats predicate and argument chunks separately and generates raw scores for each. It then combines them to compute the final similarity score. The predicate raw score is the accumulated score for all VP chunk comparisons, divided by three times the number of such comparisons. This results in a predicate raw score that is in the range [0,1], since the maximum chunk-level matching score is three. The argument raw score is produced in the same manner and multiplied by 5.0, producing a value in the range [0,5].

Where both predicate and argument raw scores exist, the total similarity score for the sentence pair is computed as the product of the two raw scores. This formulation has the benefit of permitting the degree of similarity for each score type to affect the overall score. For example, consider "Sarah bought the book," and "Sarah read the book." Here, the difference in predicate ("bought" versus "read") will temper the otherwise exact match on the arguments. Similarly, for "Sarah bought the book," and "Sarah bought the fish," the inexact match on arguments will soften the perfect predicate score.

Table 3 illustrates how the basic rule-based algorithm works. The table shows the associated chunks from each sentence, their chunk type for scoring purposes, and their chunk-level matching score values. Thus, for example, "The Korean Air deal" and "the final agreement" have a matching score value of 2 because "agreement" is a hypernym of "deal". Moreover, because there is no mention of "Bob Saling" in the first sentence, the corresponding matching value is zero.

Based on the chunk-level scores in the table, the similarity score is calculated as follows. The raw predicate score for the two predicate chunk pairs is 6 (3 for each, from the table), divided by the max-

S1: Boeing said the final agreement is expected to be signed during the next few weeks.
S2: The Korean Air deal is expected to be finalized "in the next several weeks," Boeing spokesman Bob Saling said.

S2 chunk phrase	S1 chunk phrase	Chunk type	Matching score
The Korean Air deal	the final agreement	argument	2
is expected to be finalized	is expected to be signed	predicate	3
in the next several weeks	during the next several weeks	argument	3
Boeing spokesman	Boeing	argument	3
Bob Saling		argument	0
said	said	predicate	3

Table 3. Chunk-level matching scores for rule-based scoring example from training data.

imum possible score, which is also 6, yielding a value of 1.000. The argument raw score is the sum of the scores for the argument pairs, 8 in this case, divided by the maximum possible (12 for the four argument chunks), scaled by 5, yielding a value of 3.333. The final score is their product, 3.333, which compares favorably with the gold standard score value of 3.000 for this sentence pair.

If there are no predicate chunk comparisons for the sentence pair, the rule-based scoring algorithm uses the raw argument score without modification. Similarly, where there are no argument chunk comparisons, the rule uses the raw predicate score multiplied by 5.0 to scale it to cover the range [0,5]. By being robust against zero values in this manner, the algorithm is able to handle comparisons of sentence fragments such as "Tunisia", in the event it is the entirety of the input "sentence".

Additionally, the final score that is reported is the minimum of the combined score described above and an upper limit value that is initialized at 5.0, but which can be reduced as each chunk-level comparison is performed. The upper limit value is reduced to 4.0 if there is a qualifier mismatch (e.g., "uncooked pizza" v. "pizza"). It is reduced to 3.0 if there is a number mismatch, for example, "Two men are playing chess" versus "Three men are playing chess."

5 Results and Discussion

Table 4 shows the results for both algorithms against the STS test suite. The "Corpus-based" and "Rule-based" columns reports results for the two chunk-based similarity algorithm. The five lowest rows represent the five individual data sets in the suite. The values in the table represent Pearson correlation values, which range from -1 to +1, where the closer a value is to 1, the stronger the positive correlation.

The three upper rows represent the three metrics that were used to compute global results across all of the data sets. "All" refers to the computation of a Pearson value where the five gold standards and corresponding results were concatenated. The "Allnrm" row reports correlation values obtained by scaling and translating system outputs in a manner that maintains the individual data set correlation values, yet minimizes the combined data set error. Finally, the "Mean" reports the weighted average of the individual data set correlation values, where the weights used were the numbers of sentence pairs in each data set. There were 750 sentence pairs in each of the MSRpar, MSRvid, and OnWN data sets, but only 459 in the SMT-eur data set and 399 in the SMT-news data set, for a total of 3108 sentence pairs. The characteristics of the different data sets and greater detail on the global scoring metrics are discussed further in the STS task description paper (Agirre, et al., 2012).

Category	Corpus-based	Rule-based	Improvement (%)
All	.4976	.5306	6.63%
Allnrm	.7160	.7646	6.79%
Mean	.3215	.5069	57.67%
MSRpar	.2312	.4536	96.20%
MSRvid	.6595	.7079	7.33%
SMT-eur	.1504	.3996	165.68%
On-WN	.2735	.5149	88.26%
SMT-news	.1426	.3379	136.98%

Table 4. Results against STS test suite.[1]

As Table 4 shows, the rule-based method outperformed the corpus-based method for all individual data sets and for all combined measures. The percentage improvement is noted in the rightmost column in the figure.

We believe the results for the rule-based method are sufficient to show that chunk-based methods may have a role to play in text similarity determinations, particularly in high volume applications where high throughput is essential. Chunking is computationally cheap to perform. It is also robust against sentence fragments and against incomplete or ungrammatical sentence constructions, as may be found in emails, text messages, and blog posts.

However, chunk-based methods may be restricted to such applications since, on an absolute scale, performance was in the bottom one-third of all systems that reported results against the STS data suite. Nevertheless, we recognize that our investigations into chunk-based methods were limited in both time and scope. As a result, we do not believe we have yet encountered the upper limit on performance for chunk-based text similarity systems.

[1] The results for the corpus-based chunk method are reported under the name "demetrios_glinos/task6-ATA-CHNK" on the official STS results page, http://www.cs.york.ac.uk/semeval-2012/task6/index.php?id=results-update.

References

Eneko Agirre, Daniel Cer, Mona Diab and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*.

Alberto Barrón-Cedeño, Andreas Eiselt, and Paolo Rosso. 2009. Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions. In *Proceedings of the ICON-2009: 7th International Conference on Natural Language Processing*, pp. 29-38.

Moses S. Charikar. 2002. Similarity Estimation Techniques from Rounding Algorithms. In *STOC '02 Proceedings of the thirty-fourth annual ACM symposium on theory of computing.*

Taku-ku. 2012. CRF++: Yet Another CRF toolkit. http://crfpp.googlecode.com/svn/trunk/doc/index.html.

David K. Evans, Kathleen McKeown, and Judith L. Klavans. 2005. Similarity-based Multilingual Multi-Document Summarization. *IEEE Transactions on Information Theory*.

Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. The MIT Press.

Ryan Moulton. 2010. Simple Simhashing: Clustering in linear time. [Internet]. Version 1. Ryan Moulton's Articles. Available from: http://moultano.wordpress.com/article/simple-simhashing-3kbzhsxyg4467-6/.

NLM. 2012. U.S. National Library of Medicine, Lexical Systems Group, Lexical Tools. http://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/current/web/index.html.