OPI: Semeval-2014 Task 3 System Description

Marek Kozlowski
National Information Processing Institute
mkozlowski@opi.org.pl

Abstract

In this paper, we describe the OPI system participating in the Semeval-2014 task 3 Cross-Level Semantic Similarity. Our approach is knowledge-poor, there is no exploitation of any structured knowledge resources as Wikipedia, WordNet or BabelNet. The method is also fully unsupervised, the training set is only used in order to tune the system. System measures the semantic similarity of texts using corpus-based measures of termsets similarity.

1 Introduction

The task Cross-Level Semantic Similarity of SemEval-2014 aims at an evaluation for semantic similarity across different sizes of text (lexical levels). Unlike prior SemEval tasks on textual similarity that have focused on comparing similar-sized texts, the mentioned task evaluates the case where larger text must be compared to smaller text, namely there are covered four semantic similarity comparisons: paragraph to sentence, sentence to phrase, phrase to word and word to sense.

We present the method for measuring the semantic similarity of texts using a corpus-based measure of termsets (set of words) similarity. We start from preprocessing texts, identifying boundary values, computing termsets similarities and derive from them the final score, which is normalized.

The input of the task consists of two text segments of different level. We want to determine a score indicating their semantic similarity of the smaller item to the larger item. Similarity is scored from 0 to 4, when 0 means no semantic intersection, 4 means that two items have very similar meanings.

2 Related Work

There are lots of papers about measuring the similarity between documents and single words. Document-level similarity works are based on Vector Space Models (Salton and Lesk, 1971; Salton and McGill, 1983). A significant effort has also been put into measuring similarity at the word level, namely by approaches that use distributional semantics (Turney and Pantel, 2010).

Related work can be classified into four major categories: vector-based document models methods, corpus-based methods, knowledge-based methods and hybrid methods (Islam and Inkpen, 2008).

Vector-based document models represent document as a vector of words and the similarity evaluation is based on the number of words that occur in both texts. Lexical similarity methods have problems with different words sharing common sense. Next approaches, such as corpus-based and knowledge-based methods, overcome the above issues.

Corpus based methods apply scores provided by Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA).

The Pointwise Mutual Information (PMI) (Turney, 2001) between two words w_i and w_j is:

$$PMI(w_i, w_j) = log_2 \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

The Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997; Landauer et al., 2007) is a mathematical method for modelling of the meaning of words and contexts by analysis of representative corpora. It models the meaning of words and contexts by projecting them into a vector space of reduced dimensionality, which is built up by applying singular value decomposition (SVD).

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 454–458,
Dublin, Ireland, August 23-24, 2014.

Knowledge based methods apply information from semantic networks as WordNet. They exploit the structure of WordNet to compare concepts. Leacock and Chodorow (1998) proposed metric based on the length of the shortest path between two concepts. Lesk (1986) defined similarity between concepts as the intersection between the corresponding glosses. Budanitsky and Hirst (2006) conducted the research on various WordNet-based measures. Standard thesaurus-based measures of word pair similarity are based only on a single path between concepts. By contrast Hughes and Ramage (2009) used a semantic representation of texts from random walks on WordNet.

Hybrid methods use both corpus-based measures and knowledge-based measures of word semantic similarity to determine the text similarity (Islam and Inkpen, 2008). Mihalcea and Corley (2006) suggested a combined method by exploiting corpus based measures and knowledge-based measures of words semantic similarity. Another hybrid method was proposed by Li et al. (2006) that combines semantic and syntactic information.

The methods presented above are working at fixed level of textual granularity (documents, phrases, or words). Pilehvar et al. (2013) proposed a unified approach to semantic similarity that operates at multiple levels. The method builds a common probabilistic representation over word senses in order to compare different types of linguistic data. Any lexical item is represented as a distribution over a set of word senses (obtained from WordNet), named as item's semantic signature.

3 Our Approach

Our system is fully unsupervised and knowledge-poor. It exploits Wikipedia as a raw corpus for words co-occurrence estimation. The proposed method is not using any kind of textual alignment (e.g. exploiting PoS tagging or WordNet concepts).

The method consists of four steps: preprocessing, identifying boundary values, termset-to-termset similarity computation, text-to-text similarity phase, results normalization. The results from the text-to-text similarity phase are very often beyond the range 0-4, therefore we must normalize them. We evaluated two normalization approaches: linear normalization and non-linear one. The non-linear normalization is based on built clusters (referring to integer values from 0 to 4), which are created using training data set. This step will be described in details in the section 3.5.

3.1 Preprocessing

In the first step the compared texts are retrieved, and then processed into the contexts. Context is the preprocessed original text represented as a bag of words. Texts are processed using a dictionary of proper names, name entities recognizers, PoS-taggers, providing as a result the required contexts. Contexts contain nouns, adjectives, adverbs and proper names. The output of this stage is a pair of contexts passed to the next phase.

3.2 Identifying Boundary Values

This phase is introduced in order to fast detect texts, which are unrelated (0 score) or very similar (4 score). Unrelated ones are identified basing on the lack of any co-occurrences between words from compared texts. It means that any pair of words from compared contexts do not appear together in any Wikipedia paragraph. The very similar texts are identified in two steps. At first we check if all words from the shorter texts are contained in the longer one. If the first check is not fulfilled we compute: $(c_{1,2})$ as the number of Wikipedia paragraphs that contain all of words from both contexts in the nearest neighborhood (20-words window), (c_1) and (c_2) as the numbers of Wikipedia paragraphs that contain contexts within 20-words window. If the ratio $c_{1,2}/max(c_1, c_2)$ is higher than 50% then the analyzed pair of texts refers to the same concept (very similar ones). Having two texts represented by contexts we use the proximity Lucene[1] query in order to estimate the number of Wikipedia paragraphs, which contain the words from contexts within the 20-words window.

3.3 Termset-to-termset Similarity

Termset-to-termset similarity ($t2tSim$) is defined by measure similar to PMI. Given a dictionary D and two termsets (set of words) $W_i \subseteq D$ and $W_j \subseteq D$ then the measure is expressed by the formula:

$$t2tSim(W_i, W_j) = \frac{c(W_i, W_j)}{min(c(W_i), c(W_j))}$$

Here, $c(X_1, .., X_n)$ is a number of Wikipedia paragraphs that contain all terms covered by termsets

[1] http://lucene.apache.org/core/

$X_1, .., X_n$. Two input termsets are semantically close if the similarity measure $t2tSim$ is higher than the user-defined threshold (e.g. 10%). Comparing to the previous step we use the minimum operator in the formula's denominator in order to take into account even one directed relevant association. It was proved experimentally that the proposed measure leads to better results than the PMI measure using NEAR query (co-occurrence within a 10-words window). Specifically, the following formula is used to collect the PMI value between termsets using the Wikipedia as a background corpus:

$$PMI(W_i, W_j) = log_2 \frac{c(W_i, W_j) * WikiSize}{c(W_i) * c(W_j)}$$

In the performed experiments we approximated the value of $WikiSize$ to 30 millions (number of paragraphs of English articles in Wikipedia). In table 1 we present results of Spearman correlation reported by the System using different measures PMI and $t2tSim$. The second measure is slightly better therefore it was chosen as the final one. These correlations were computed after linear normalization of the output measures.

Level	Measure	Spearman correlation
word2sense	PMI	19
word2sense	t2tSim	19
phrase2word	PMI	29
phrase2word	t2tSim	29
sentence2phrase	PMI	45
sentence2phrase	t2tSim	47
paragraph2sentence	PMI	48
paragraph2sentence	t2tSim	49

Table 1: Comparison of PMI and t2tSim measures in the semantic similarity task using Spearman correlation (percentages).

3.4 Text-to-text Similarity

Given two input texts we compute the termset-to-termset similarities in order to derive the final semantic score. We attempt to model the semantic similarity of texts as a function of the semantic similarities of the component termsets. We do this by combining metrics of termset-to-termset similarities and weights into a formula that is a potentially good indicator of semantic similarity of the two input texts. Weights ($w_{m1} > w_{m2} > w_{m3}$) are experimentally set with linear scalable values $w_{m1} = 4, w_{m2} = 2, w_{m3} = 1$ respectively. The pseudo-code of this phase is in Algorithm 1.

Algorithm 1 Text-to-text similarity

Input: c_s, c_l are contexts representing shorter and longer texts respectively; w_{m1}, w_{m2}, w_{m3} as weights for different scopes of similarity comparison;

Output: m as a similarity measure

```
m = 0
m = m + t2tSim(c_s, c_l) * w_m1
for term t_i ∈ c_l do
    m = m + t2tSim(c_s, {t_i}) * w_m2
end for
for term t_j ∈ c_s do
    m = m + t2tSim(c_l, {t_j}) * w_m2
end for
for term t_i ∈ c_s do
    for term t_j ∈ c_l do
        m = m + t2tSim({t_i}, {t_j}) * w_m3
    end for
end for
return m
```

3.5 Results Normalization

The crucial part of the method is a process of normalization obtained measures into the range (0,4). The values 0 and 4 are covered by the step described in the section 3.2. We need to normalize values from the text-to-text similarity phase. This step can be done in two ways: linear normalization and non-linear one. The first one is a casual transformation defined as dividing elements by theirs maximum and scaling to 4. The second one is based on clustering training set. In other words, using training set we induce rules how reported text-to-text similarity values should be transformed into the range (0,4). We implemented hierarchical agglomerative clustering algorithm (with average linkage)[2] in order to cluster similarity measures into five distinct groups. Sorted centroids of the above created groups are labeled with values 0 to 4 respectively. For each new similarity measure (obtained in the testing phase) we measure the distance to the closest cluster's centroids. The final value is derived linearly

[2]Hierarchical Agglomerative Clustering treats initially each instance as a singleton cluster and then successively agglomerate pairs of clusters using the average distance between cluster's elements until the user defined number of clusters persist.

from the distance to the centroids (i.e. if the value is in the middle between centroids referring to 1 and 2, we assign as a final value 1.5). In the testing step we use the non-linear normalization, the evaluations on training set show that clustering based approach provides marginal improvement against linear normalization (about 1% according to Spearman rank, 4-8% according to Pearson correlation).

4 Results

In Task 3, systems were evaluated both within one of four comparison types and also across all comparison types. The system outputs and gold standard ratings are compared in two ways, using Pearson correlation and Spearman's rank correlation (rho). Pearson correlation tests the degree of similarity between the system's similarity ratings and the gold standard ratings. Spearman's rho tests the degree of similarity between the rankings of the items according to similarity. Ranks were computed by summing the correlation values across all four levels of comparisons. The sum of the Pearson correlations is used for the official rank of Task 3. However, the organizers provide a second ranking using the sum of the Spearman correlations.

Level	System	Pearson/ Spearman
word2sense	OPI	15.2/13.1
word2sense	SimCompass	35.6/34.4
word2sense	Baseline	10.9/13.0
phrase2word	OPI	21.3/18.8
phrase2word	SimCompass	41.5/42.4
phrase2word	Baseline	16.5/16.2
sentence2phrase	OPI	43.3/42.4
sentence2phrase	SimCompass	74.2/72.8
sentence2phrase	Baseline	56.2/62.6

Table 2: Results for Pearson and Spearman correlation (percentages) scored by OPI System, SimCompass (the best performing one) and the Baseline one.

We submitted only one run in three comparison types. We avoided the paragraph-to-sentence comparison. Evaluations on training set show that our method reports values below the baseline in both types: paragraph-to-sentence and sentence-to-phrase. In the testing phase we decided to perform only sentence-to-phrase comparison because it reports better values than paragraph-to-sentence according to Pearson correlation, which is used for the official rank.

The best results our algorithm scores in the category phrase-to-word. In this comparison type it was ranked at 12th position among 21 participating systems. In the word-to-sense it was at 14th position among 20 systems. The word-to-sense comparison is converted into the task similar to phrase-to-word by using glosses of target senses. Each key of WordNet sense is replaced with its gloss. It is the only situation when we use the external knowledge resources, but it is not a part of the algorithm. The last comparison (sentence-to-phrase) was our worst, because we did not beat the baseline, as we did in the previous categories. In the sentence-to-phrase comparison word alignment or syntax parsing seems to be very important, in our case none of them was applied. The main conclusion is that comparison of larger text units can not be based on bag of words approaches, where order of words is not important. Let us recall that our method is knowledge-poor, what leads to difficulties in evaluating it against knowledge-rich ones (using sense inventories e.g. WordNet). Generally, we scored better results using Pearson correlation than Spearman's one.

5 Conclusions

We presents our cross-level semantic similarity method, which is knowledge-poor (not using any kind of structured information from resources like machine-readable dictionaries, thesaurus, or ontologies) and fully unsupervised (there is no learning phase leading to models enable to categorize compared texts). The method exploits only Wikipedia as a raw corpora in order to estimate frequencies of co-occurrences. We were aimed to verify how good results can be achieved using only corpus-based approach and not including algorithms that have embedded deep language knowledge. The system scores best in the phrase-to-word (12th rank) and word-to-sense (14th rank) types of comparison with regard to Pearson correlation, while performing a little worse with the Spearman's correlation. The worst results were reported in the sentence-to-phrase category, which brings us the conclusion that larger text units demand word alignment, syntax parsing and more sophisticated text-to-text similarity models.

References

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1): 13–47.

Thomas Hughes and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walk. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 581–589.

Aminul Islam and Diana Inkpen. 2008. Semantic Text Similarity using Corpus-Based Word Similarity and String Similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2): 1–25.

Thomas Landauer and Susan Dumais. 1997. A solution to Platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104: 211–240.

Thomas Landauer, Danielle McNamara, Simon Dennis and Walter Kintsch. 2007. *Handbook of Latent Semantic Analysis*. Psychology Press.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*, pages 265–283.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference*, pages 24–26.

Yuhua Li, David McLean, Zuhair Bandar, James O'Shea and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8): 1138-1149.

Rada Mihalcea, Courtney Corley and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of the American Association for Artificial Intelligence*, pages 775–780.

Mohammad Pilehvar, David Jurgens and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351.

Gerard Salton and Michael Lesk. 1971. *Computer evaluation of indexing and text processing*. Prentice-Hall, Englewood Cliffs, New Jersey.

Gerard Salton and Michael McGill. 1983. Alternation. *Introduction to modern information retrieval*. McGraw-Hill.

Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Articial Intelligence Research*, 37: 141–188.

Peter Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*, pages 491–502.