ECNUCS: Measuring Short Text Semantic Equivalence Using Multiple Similarity Measurements

Tian Tian ZHU
Department of Computer Science and Technology
East China Normal University
51111201046@student.ecnu.edu.cn

Man LAN*
Department of Computer Science and Technology
East China Normal University
mlan@cs.ecnu.edu.cn

Abstract

This paper reports our submissions to the Semantic Textual Similarity (STS) task in *SEM Shared Task 2013. We submitted three Support Vector Regression (SVR) systems in core task, using 6 types of similarity measures, i.e., string similarity, number similarity, knowledge-based similarity, corpus-based similarity, syntactic dependency similarity and machine translation similarity. Our third system with different training data and different feature sets for each test data set performs the best and ranks 35 out of 90 runs. We also submitted two systems in typed task using string based measure and Named Entity based measure. Our best system ranks 5 out of 15 runs.

1 Introduction

The task of semantic textual similarity (STS) is to measure the degree of semantic equivalence between two sentences, which plays an increasingly important role in natural language processing (NLP) applications. For example, in text categorization (Yang and Wen, 2007), two documents which are more similar are more likely to be grouped in the same class. In information retrieval (Sahami and Heilman, 2006), text similarity improves the effectiveness of a semantic search engine by providing information which holds high similarity with the input query. In machine translation (Kauchak and Barzilay, 2006), sentence similarity can be applied for automatic evaluation of the output translation and the reference translations. In question answering (Mohler and Mihalcea, 2009), once the question and the candidate answers are treated as two texts, the answer text which has a higher relevance with the question text may have higher probability to be the right one.

The STS task in *SEM Shared Task 2013 consists of two subtasks, i.e., core task and typed task, and we participate in both of them. The core task aims to measure the semantic similarity of two sentences, resulting in a similarity score which ranges from 5 (semantic equivalence) to 0 (no relation). The typed task is a pilot task on typed-similarity between semi-structured records. The types of similarity to be measured include location, author, people involved, time, events or actions, subject and description as well as the general similarity of two texts (Agirre et al., 2013).

In this work we present a Support Vector Regression (SVR) system to measure sentence semantic similarity by integrating multiple measurements, i.e., string similarity, knowledge based similarity, corpus based similarity, number similarity and machine translation metrics. Most of these similarities are borrowed from previous work, e.g., (Bär et al., 2012), (Šaric et al., 2012) and (de Souza et al., 2012). We also propose a novel syntactic dependency similarity. Our best system ranks 35 out of 90 runs in core task and ranks 5 out of 15 runs in typed task.

The rest of this paper is organized as follows. Section 2 describes the similarity measurements used in this work in detail. Section 3 presents experiments and the results of two tasks. Conclusions and future work are given in Section 4.

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 124–131, Atlanta, Georgia, June 13-14, 2013.

2 Text Similarity Measurements

To compute semantic textual similarity, previous work has adopted multiple semantic similarity measurements. In this work, we adopt 6 types of measures, i.e., string similarity, number similarity, knowledge-based similarity, corpus-based similarity, syntactic dependency similarity and machine translation similarity. Most of them are borrowed from previous work due to their superior performance reported. Besides, we also propose two syntactic dependency similarity measures. Totally we get 33 similarity measures. Generally, these similarity measures are represented as numerical values and combined using regression model.

2.1 Preprocessing

Generally, we perform text preprocessing before we compute each text similarity measurement. Firstly, Stanford parser[1] is used for sentence tokenization and parsing. Specifically, the tokens *n't* and *'m* are replaced with *not* and *am*. Secondly, Stanford POS Tagger[2] is used for POS tagging. Thirdly, Natural Language Toolkit[3] is used for WordNet based Lemmatization, which lemmatizes the word to its nearest base form that appears in WordNet, for example, was is lemmatized as is, not be.

Given two short texts or sentences s_1 and s_2, we denote the word set of s_1 and s_2 as S_1 and S_2, the length (i.e., number of words) of s_1 and s_2 as $|S_1|$ and $|S_2|$.

2.2 String Similarity

Intuitively, if two sentences share more strings, they are considered to have higher semantic similarity. Therefore, we create 12 string based features in consideration of the common sequence shared by two texts.

Longest Common sequence (LCS). The widely used *LCS* is proposed by (Allison and Dix, 1986), which is to find the maximum length of a common subsequence of two strings and here the subsequence need to be contiguous. In consideration of the different length of two texts, we compute *LCS* similarity using Formula (1) as follows:

$$Sim_{LCS} = \frac{Length\_of\_LCS}{min(|S_1|, |S_2|)} \quad (1)$$

In order to eliminate the impacts of various forms of word, we also compute a *Lemma LCS* similarity score after sentences being lemmatized.

word n-grams. Following (Lyon et al., 2001), we calculate the *word n-grams* similarity using the Jaccard coefficient as shown in Formula (2), where p is the number of n-grams shared by s_1 and s_2, q and r are the number of n-grams not shared by s_1 and s_2, respectively.

$$Jacc = \frac{p}{p+q+r} \quad (2)$$

Since we focus on short texts, here only n=1,2,3,4 is used in this work. Similar with *LCS*, we also compute a *Lemma n-grams* similarity score.

Weighted Word Overlap (WWO). (Šaric et al., 2012) pointed out that when measuring sentence similarity, different words may convey different content information. Therefore, we consider to assign more importance to those words bearing more content information. To measure the importance of each word, we use Formula (3) to calculate the information content for each word w:

$$ic(w) = \ln \frac{\sum_{w' \in C} freq(w')}{freq(w)} \quad (3)$$

where C is the set of words in the corpus and $freq(w)$ is the frequency of the word w in the corpus. To compute $ic(w)$, we use the Web 1T 5-gram Corpus[4], which is generated from approximately one trillion word tokens of text from Web pages.

Obviously, the *WWO* scores between two sentences is non-symmetric. The *WWO* of s_2 by s_1 is given by Formula (4):

$$Sim_{wwo}(s_1, s_2) = \frac{\sum_{w \in S_1 \cap S_2} ic(w)}{\sum_{w' \in S_2} ic(w')} \quad (4)$$

Likewise, we can get $Sim_{wwo}(s_2, s_1)$ score. Then the final *WWO* score is the harmonic mean of $Sim_{wwo}(s_1, s_2)$ and $Sim_{wwo}(s_2, s_1)$. Similarly, we get a *Lemma WWO* score as well.

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://nlp.stanford.edu/software/tagger.shtml
[3] http://nltk.org/
[4] http://www.ldc.upenn.edu/Catalog/docs/LDC2006T13

2.3 Knowledge Based Similarity

Knowledge based similarity approaches rely on a semantic network of words. In this work all knowledge-based word similarity measures are computed based on WordNet. For word similarity, we employ four WordNet-based similarity metrics: the ***Path*** similarity (Banea et al., 2012); the ***WUP*** similarity (Wu and Palmer, 1994); the ***LCH*** similarity (Leacock and Chodorow, 1998); the ***Lin*** similarity (Lin, 1998). We adopt the NLTK library (Bird, 2006) to compute all these word similarities.

In order to determine the similarity of sentences, we employ two strategies to convert the word similarity into sentence similarity, i.e., (1) the best alignment strategy (*align*) (Banea et al., 2012) and (2) the aggregation strategy (*agg*) (Mihalcea et al., 2006).

The best alignment strategy is computed as below:

$$Sim_{align}(s_1, s_2) = \frac{(\omega + \sum_{i=1}^{|\varphi|} \varphi_i) * (2|S_1||S_2|)}{|S_1| + |S_2|} \quad (5)$$

where ω is the number of shared terms between s_1 and s_2, list φ contains the similarities of non-shared words in shorter text, φ_i is the highest similarity score of the ith word among all words of the longer text. The aggregation strategy is calculated as below:

$$Sim_{agg}(s_1, s_2) = \frac{\sum_{w \in S_1}(maxSim(w, S_2) * ic(w))}{\sum_{w \in \{S_1\}} ic(w)} \quad (6)$$

where $maxSim(w, S_2)$ is the highest WordNet-based score between word w and all words of sentence S_2. To compute $ic(w)$, we use the same corpus as *WWO*, i.e., the Web 1T 5-gram Corpus. The final score of the aggregation strategy is the mean of $Sim_{agg}(s_1, s_2)$ and $Sim_{agg}(s_2, s_1)$. Finally we get 8 knowledge based features.

2.4 Corpus Based Similarity

Latent Semantic Analysis (LSA) (Landauer et al., 1997). In LSA, term-context associations are captured by means of a dimensionality reduction operation performing singular value decomposition (SVD) on the term-by-context matrix T, where T is induced from a large corpus. We use the TASA corpus[5] to obtain the matrix and compute the word similarity using *cosine* similarity of the two vectors of the words. After that we transform word similarity to sentence similarity based on Formula (5).

Co-occurrence Retrieval Model (CRM) (Weeds, 2003). *CRM* is based on a notion of substitutability. That is, the more appropriate it is to substitute word w_1 in place of word w_2 in a suitable natural language task, the more semantically similar they are. The degree of substitutability of w_2 with w_1 is dependent on the proportion of co-occurrences of w_1 that are also the co-occurrences of w_2, and the proportion of co-occurrences of w_2 that are also the co-occurrences of w_1. Following (Weeds, 2003), the *CRM* word similarity is computed using Formula (7):

$$Sim_{CRM}(w_1, w_2) = \frac{2 * |c(w_1) \cap c(w_2)|}{|c(w_1)| + |c(w_2)|} \quad (7)$$

where $c(w)$ is the set of words that co-occur with w. We use the 5-gram part of the Web 1T 5-gram Corpus to obtain $c(w)$. If two words appear in one 5-gram, we will treat one word as the co-occurring word of each other. To obtain $c(w)$, we propose two methods. In the first *CRM* similarity, we only consider the word w with $|c(w)| > 200$, and then take the top 200 co-occurring words ranked by the co-occurrence frequency as its $c(w)$. To relax restrictions, we also present an extended *CRM* (denoted by *ExCRM*), which extends the *CRM* list that all w with $|c(w)| > 50$ are taken into consideration, but the maximum of $|c(w)|$ is still set to 200. Finally, these two *CRM* word similarity measures are transformed to sentence similarity using Formula (5).

2.5 Syntactic Dependency Similarity

As (Šaric et al., 2012) pointed out that dependency relations of sentences often contain semantic information, in this work we propose two novel syntactic dependency similarity features to capture their possible semantic similarity.

Simple Dependency Overlap. First we measure the simple dependency overlap between two sentences based on matching dependency relations. Stanford Parser provides 53 dependency relations, for example:

$$nsubj(remain - 16, leader - 4)$$
$$dobj(return - 10, home - 11)$$

[5] http://lsa.colorado.edu/

where $nsubj$ (nominal subject) and $dobj$ (direct object) are two dependency types, $remain$ is the governing lemma and $leader$ is the dependent lemma. Two syntactic dependencies are considered equal when they have the same dependency type, governing lemma, and dependent lemma.

Let R_1 and R_2 be the set of all dependency relations in s_1 and s_2, we compute Simple Dependency Overlap using Formula (8):

$$Sim_{SimDep}(s_1, s_2) = \frac{2 * |R_1 \cap R_2| * |R_1||R_2|}{|R_1| + |R_2|} \quad (8)$$

Special Dependency Overlap. Several types of dependency relations are believed to contain the primary content of a sentence. So we extract three roles from those special dependency relations, i.e., *predicate*, *subject* and *object*. For example, from above dependency relation $dobj$, we can extract the object of the sentence, i.e., *home*. For each of these three roles, we get a similarity score. For example, to calculate $Sim_{predicate}$, we denote the sets of predicates of two sentences as S_{p1} and S_{p2}. We first use *LCH* to compute word similarity and then compute sentence similarity using Formula (5). Similarly, the Sim_{subj} and Sim_{obj} are obtained in the same way. In the end we average the similarity scores of the three roles as the final Special Dependency Overlap score.

2.6 Number Similarity

Numbers in the sentence occasionally carry similarity information. If two sentences contain different sets of numbers even though their sentence structure is quite similar, they may be given a low similarity score. Here we adopt two features following (Šaric et al., 2012), which are computed as follow:

$$\log(1 + |N_1| + |N_2|) \quad (9)$$

$$2 * |N_1 \cap N_2|/(|N_1| + |N_2|) \quad (10)$$

where N_1 and N_2 are the sets of all numbers in s_1 and s_2. We extract the number information from sentences by checking if the POS tag is *CD* (cardinal number).

2.7 Machine Translation Similarity

Machine translation (MT) evaluation metrics are designed to assess whether the output of a MT system is semantically equivalent to a set of reference translations. The two given sentences can be viewed as one input and one output of a MT system, then the MT measures can be used to measure their semantic similarity. We use the following 6 lexical level metrics (de Souza et al., 2012): *WER*, *TER*, *PER*, *NIST*, *ROUGE-L*, *GTM-1*. All these measures are obtained using the Asiya Open Toolkit for Automatic Machine Translation (Meta-) Evaluation[6].

3 Experiment and Results

3.1 Regression Model

We adopt LIBSVM[7] to build Support Vector Regression (SVR) model for regression. To obtain the optimal SVR parameters C, g, and p, we employ grid search with 10-fold cross validation on training data. Specifically, if the score returned by the regression model is bigger than 5 or less than 0, we normalize it as 5 or 0, respectively.

3.2 Core Task

The organizers provided four different test sets to evaluate the performance of the submitted systems. We have submitted three systems for core task, i.e., Run 1, Run 2 and Run 3. Run 1 is trained on all training data sets with all features except the number based features, because most of the test data do not contain number. Run 2 uses the same feature sets as Run 1 but different training data sets for different test data as listed in Table 1, where different training data sets are combined together as they have similar structures with the test data. Run 3 uses different feature sets as well as different training data sets for each test data. Table 2 shows the best feature sets used for each test data set, where "+" means the feature is selected and "-" means not selected. We did not use the whole feature set because in our preliminary experiments, some features performed not well on some training data sets, and they even reduced the performance of our system. To select features, we trained two SVR models for each feature, one with all features and another with all features except this feature. If the first model outperforms the second model, this feature is chosen.

Table 3 lists the performance of these three systems as well as the baseline and the best results on

[6]http://nlp.lsi.upc.edu/asiya/

[7]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Test	Training
Headline	MSRpar
OnWN+FNWN	MSRpar+OnWN
SMT	SMTnews+SMTeuroparl

Table 1: Different training data sets used for each test data set

type	Features	Headline	OnWN and FNWN	SMT
String Based	LCS	+	+	-
	Lemma LCS	+	+	-
	N-gram	+	1+2gram	1gram
	Lemma N-gram	+	1+2gram	1gram
	WWO	+	+	+
	Lemma WWO	+	+	+
Knowledge Based	Path,WUP,LCH,Lin +aligh	+	+	+
	Path,WUP,LCH,Lin +ic-weighted	+	+	+
Corpus Based	LSA	+	+	+
	CRM,ExCRM	+	+	+
Syntactic Dependency	Simple Dependency Overlap	+	+	+
	Special Dependency Overlap	+	-	+
Number	Number	+	-	-
MT	WER	-	+	+
	TER	-	+	+
	PER	+	+	+
	NIST	+	+	-
	ROUGE-L	+	+	+
	GTM-1	+	+	+

Table 2: Best feature combination for each data set

System	Mean	Headline	OnWN	FNWN	SMT
Best	0.6181	0.7642	0.7529	0.5818	0.3804
Baseline	0.3639	0.5399	0.2828	0.2146	0.2861
Run 1	0.3533	0.5656	0.2083	0.1725	0.2949
Run 2	0.4720	0.7120	0.5388	0.2013	0.2504
Run 3 (rank 35)	0.4967	0.6799	0.5284	0.2203	0.3595

Table 3: Final results on STS core task

STS core task in *SEM Shared Task 2013. For the three runs we submitted to the task organizers, Run 3 performs the best results and ranks 35 out of 90 runs. Run 2 performs much better than Run 1. It indicates that using different training data sets for different test sets indeed improves results. Run 3 outperforms Run 2 and Run 1. It shows that our feature selection process for each test data set does help im-

prove the performance too. From this table, we find that different features perform different on different kinds of data sets and thus using proper feature subsets for each test data set would make improvement.

Besides, results on the four test data sets are quite different. *Headline* always gets the best result on each run and *OnWN* follows second. And results of *FNWN* and *SMT* are much lower than *Headline* and *OnWN*. One reason of the poor performance of *FNWN* may be the big length difference of sentence pairs. That is, sentence from WordNet is short while sentence from FrameNet is quite longer, and some samples even have more than one sentence (e.g. "*doing as one pleases or chooses*" VS "*there exist a number of different possible events that may happen in the future in most cases, there is an agent involved who has to consider which of the possible events will or should occur a salient entity which is deeply involved in the event may also be mentioned*"). As a result, even though the two sentences are similar in meaning, most of our measures would give low scores due to quite different sentence length.

In order to understand the contributions of each similarity measurement, we trained 6 SVR regression models based on 6 types on *MSRpar* data set. Table 4 presents the Pearson's correlation scores of the 6 types of measurements on *MSRpar*. We can see that the corpus-based measure achieves the best, then the knowledge-based measure and the MT measure follow. Number similarity performs surprisingly well, which benefits from the property of data set that *MSRpar* contains many numbers in sentences and the sentence similarity depends a lot on those numbers as well. The string similarity is not as good as the knowledge-based, the corpus-based and the MT similarity because of its disability of extracting semantic characteristics of sentence. Surprisingly, the Syntactic dependency similarity performs the worst. Since we only extract two features based on sentence dependency, they may not enough to capture the key semantic similarity information from the sentences.

3.3 Typed Task

For typed task, we also adopt a SVR model for each type. Since several previous similarity measures used for core task are not suitable for evaluation of the similarity of *people involved*, *time period*, *location* and *event or action involved*, we add two Named Entity Recognition (NER) based features. Firstly we use Stanford NER[8] to obtain person, location and date information from the whole text with NER tags of "PERSON", "LOCATION" and "DATE". Then for each list of entity, we get two feature values using the following two formulas:

Features	results
string	0.4757
knowledge-based	**0.5640**
corpus-based	**0.5842**
syntactic dependency	0.3528
number	**0.5278**
MT metrics	**0.5595**

Table 4: Pearson correlation of features of the six aspects on MSRpar

$$Sim_{NER\_Num}(L1_{NER}, L2_{NER}) = \frac{min(|L1_{NER}|, |L2_{NER}|)}{max(|L1_{NER}|, |L2_{NER}|)} \quad (11)$$

$$Sim_{NER}(L1_{NER}, L2_{NER}) = \frac{Num(equalpairs)}{|L1_{NER}| * |L2_{NER}|} \quad (12)$$

where L_{NER} is the list of one entity type from the text, and for two lists of NERs $L1_{NER}$ and $L2_{NER}$, there are $|L1_{NER}| * |L2_{NER}|$ NER pairs. $Num(equalpairs)$ is the number of equal pairs. Here we expand the condition of equivalence: two NERs are considered equal if one is part of another (e.g. "John Warson" VS "Warson"). Features and content we used for each similarity are presented in Table 5. For the three similarities: *people involved*, *time period*, *location*, we compute the two NER based features for each similarity with NER type of "PERSON", "LOCATION" and "DATE". And for *event or action involved*, we add the above 6 NER feature scores as its feature set. The NER based similarity used in *description* is the same as *event or action involved* but only based on "dcDescription" part of text. Besides, we add a length feature in *description*, which is the ratio of shorter length and longer length of descriptions.

[8] http://nlp.stanford.edu/software/CRF-NER.shtml

Type	Features	Content used
author	string based (+ knowledge based for Run2)	dcCreator
people involved	NER based	whole text
time period	NER based	whole text
location	NER based	whole text
event or action involved	NER based	whole text
subject	string based (+ knowledge based for Run2)	dcSubject
description	string based, NER based,length	dcDescription
General	the 7 similarities above	

Table 5: Feature sets and content used of 8 type similarities of Typed data

We have submitted two runs. Run 1 uses only string based and NER based features. Besides features used in Run 1, Run 2 also adds knowledge based features. Table 6 shows the performance of our two runs as well as the baseline and the best results on STS typed task in *SEM Shared Task 2013. Our Run 1 ranks 5 and Run 2 ranks 7 out of 15 runs. Run 2 performed worse than Run 1 and the possible reason may be the knowledge based method is not suitable for this kind of data. Furthermore, since we only use NER based features which involves three entities for these similarities, they are not enough to capture the relevant information for other types.

4 Conclusion

In this paper we described our submissions to the Semantic Textual Similarity Task in *SEM Shared Task 2013. For core task, we collect 6 types of similarity measures, i.e., string similarity, number similarity, knowledge-based similarity, corpus-based similarity, syntactic dependency similarity and machine translation similarity. And our Run 3 with different training data and different feature sets for each test data set ranks 35 out of 90 runs. For typed task, we adopt string based measure, NER based measure and knowledge based measure, our best system ranks 5 out of 15 runs. Clearly, these similarity measures are not quite enough. For the core task, in our future work we will consider the measures to evaluate the sentence difference as well. For the typed task, with the help of more advanced IE tools to extract more information regarding different types, we need to propose more methods to evaluate the similarity.

Acknowledgments

The authors would like to thank the organizers and reviewers for this interesting task and their helpful suggestions and comments, which improved the final version of this paper. This research is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Pujiang Talent Program (No.09PJ1404500), Doctoral Fund of Ministry of Education of China (No.20090076120029) and Shanghai Knowledge Service Platform Project (No.ZF1213).

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

Lloyd Allison and Trevor I Dix. 1986. A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23(5):305–310.

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. pages 635–642. First Joint Conference on Lexical and Computational Semantics (*SEM).

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. pages 435–440. First Joint Conference on Lexical and Computational Semantics (*SEM).

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

System	general	author	people	time	location	event	subject	description	mean
Best	0.7981	0.8158	0.6922	0.7471	0.7723	0.6835	0.7875	0.7996	0.7620
Baseline	0.6691	0.4278	0.4460	0.5002	0.4835	0.3062	0.5015	0.5810	0.4894
Run 1	0.6040	0.7362	0.3663	0.4685	0.3844	0.4057	0.5229	0.6027	0.5113
Run 2	0.6064	0.5684	0.3663	0.4685	0.3844	0.4057	0.5563	0.6027	0.4948

Table 6: Final results on STS typed task

José Guilherme C de Souza, Matteo Negri, Trento Povo, and Yashar Mehdad. 2012. Fbk: Machine translation evaluation and word similarity metrics for semantic textual similarity. pages 624–630. First Joint Conference on Lexical and Computational Semantics (*SEM).

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 455–462. Association for Computational Linguistics.

Thomas K Landauer, Darrell Laham, Bob Rehder, and Missy E Schreiner. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th international conference on Machine Learning*, volume 1, pages 296–304. San Francisco.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 118–125.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the national conference on artificial intelligence*, volume 21, page 775. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 567–575. Association for Computational Linguistics.

Mehran Sahami and Timothy D Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. ACM.

Frane Šaric, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašic. 2012. Takelab: Systems for measuring semantic text similarity. pages 441–448. First Joint Conference on Lexical and Computational Semantics (*SEM).

Julie Elizabeth Weeds. 2003. *Measures and applications of lexical distributional similarity*. Ph.D. thesis, Citeseer.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Cha Yang and Jun Wen. 2007. Text categorization based on similarity approach. In *Proceedings of International Conference on Intelligence Systems and Knowledge Engineering (ISKE)*.