TATO: Leveraging on Multiple Strategies for Semantic Textual Similarity

Tu Thanh Vu[†], **Quan Hung Tran**[††], **Son Bao Pham**[†]
[†]University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[††]Japan Advanced Institute of Science and Technology, Japan
[†] {tuvt,sonpb}@vnu.edu.vn
[††] quanth@jaist.ac.jp

Abstract

In this paper, we describe the TATO system which participated in the SemEval-2015 Task 2a: "Semantic Textual Similarity (STS) for English". Our system is trained on published datasets from the previous competitions. Based on some machine learning techniques, it combines multiple similarity measures of varying complexity ranging from simple lexical and syntactic similarity measures to complex semantic similarity ones to compute semantic textual similarity. Our final model consists of a simple linear combination of about 30 main features out of a numerous number of features experimented. The results are promising, with Pearson's coefficients on each individual dataset ranging from 0.6796 to 0.8167 and an overall weighted mean score of 0.7422, well above the task baseline system.

1 Introduction

Measuring semantic textual similarity (STS) can be defined as the task of computing the degree of semantic equivalence between pairs of texts. It has drawn an increasing amount of attention from the NLP community, especially at level of short text fragments, as partly reflected in the SemEval tasks in recent years. In the SemEval-2015 Task 2, the degree of semantic equivalence for each sentence pair is represented by a similarity score between 0 (no relation) and 5 (semantic equivalence). STS has a wide range of applications which includes applications for machine translation evaluation, information extraction, question answering, and summarization.

STS is related to, but different from textual entailment (TE) (Dagan et al., 2006) and paraphrase recognition (PARA) (Dolan et al., 2004) as it aims to render a graded notion of semantic equivalence between two textual snippets, rather than a binary yes/no decision. STS requires a bidirectional similarity relation between sentences, while TE annotates them with an unidirectional entailment relation.

The literature of STS is rife with attempts to compute similarity between texts using a multitude of measures at different levels of depth: lexical (Malakasiotis and Androutsopoulos, 2007), syntactic (Malakasiotis, 2009; Zanzotto et al., 2009), and semantic (Rinaldi et al., 2003; Bos and Markert, 2005). (Gomaa and Fahmy, 2013) discusses existing works on STS and partitions them into three categories based on the similarity measures used: *(i)* string-based approaches (Bär et al., 2012; Malakasiotis and Androutsopoulos, 2007) which operate on string sequences and character composition to compute similarities and can be categorized into two groups: character-based and term-based approaches; *(ii)* corpus-based approaches (Li et al., 2006) which gain statistics information about words from large corpora and reflect their semantics in distributional high semantic space to determine the similarity, such as Latent Semantic Analysis (LSA) (Landauer et al., 1998; Foltz et al., 1998) and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007); *(iii)* knowledge-based approaches (Mihalcea et al., 2006) which determine the degree of similarity between texts using information derived from semantic networks, such as WordNet (Miller, 1995).

Though each of these existing measures has its own advantages, they are typically used in separation. In our work, we integrate multiple similarity

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 190–195,
Denver, Colorado, June 4-5, 2015.

measures of varying complexity ranging from simple lexical and syntactic similarity measures to complex semantic similarity ones and rely on supervised machine learning to take advantage of the different contributions of different features.

We organize the remainder of the paper as follows: Section 2 describes the features in detail. Section 3 presents the machine learning setup and our submitted system. Sections 4 discusses the results. The conclusions follow in the final section.

2 Text Similarity Measures

In this section, we describe the various features we experimented and selected for our final model.

2.1 Lexical Similarity Measures

2.1.1 Word/Phrase Alignment Measures

When two sentences are related semantically, they tend to be similar in appearance. Hence, we develop an automatic word/phrase alignment module based on the METEOR metric (Denkowski and Lavie, 2010) to align corresponding words and phrases between each pair of sentences. Alignments here are based on exact, stem, synonym (via WordNet), and paraphrase (via a lookup table) matches between words and phrases. Given two sentences of text, s_1 and s_2 (stop-words are removed from each sentence), we define the following metrics:

$$\mathscr{I}(s_1,s_2) = \left|numOfMatches(s_1,s_2) - \frac{min\{len(s_1),len(s_2)\}}{2}\right|$$

and

$$\mathscr{D}(s_1,s_2) = \frac{2\times numOfMatches(s_1,s_2)}{min\{len(s_1),len(s_2)\}},$$

where $numOfMatches(s_1, s_2)$ and $len(s)$ are the number of aligned word/phrase pairs between s_1 and s_2, and the number of words in s, respectively.

2.1.2 Machine Translation Measures

We treat the task as a monolingual machine translation (MT) task (the source and target languages are the same, and the input and output should be similar in meaning), and take advantage of a variety of MT measures. At the lexical level, we experiment different n-gram and edit-distance-based metrics.

BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and METEOR (Denkowski and Lavie, 2010) are n-gram-based metrics commonly used for MT evaluation. BLEU scores the target output by counting n-gram matches with the reference, relying on exact matching and has no concept of synonymy or paraphrasing. NIST is similar to BLEU, however, it uses the arithmetic mean of n-gram overlaps, rather than the geometric mean. Unlike BLEU which focuses on precision, METEOR uses a combination of both precision and recall. Moreover, it incorporates stemming, synonymy and paraphrase. MAXSIM (Chan and Ng, 2008) models the MT problem as a maximum bipartite matching one and maps each word in one sentence to at most one word in the other sentence. We also experiment with TESLA (Liu et al., 2010) - a variant of MAXSIM.

Besides those, we also use edit-distance-based metrics. TER (Snover et al., 2006) and TERp (Snover et al., 2009) measure the number of edit operations (e.g. insertions, deletions, and substitutions) necessary to transform one text into the other.

2.2 Syntactic Similarity Measures

2.2.1 Content Word Match and Mismatch

Given a sentence pair, we extract corresponding content words (nouns, verbs, adjectives, and adverbs) between the sentences. This syntactic information is obtained from the Stanford parser (Klein and Manning, 2003). We have both the proportions of aligned words and the proportions of unaligned words in the two sentences (by normalizing with the harmonic mean of their number of content words) for each lexical category of content word.

2.2.2 Subject-Verb-Object Comparison

We also employ dependency parsing in measuring semantic similarity. Specifically, some attributes like subjects, verbs, objects are identified for each pair of sentences. These attributes are used for our matching procedure which is based on the following comparisons between each pair of sentences:

- Subject-Subject Comparison
- Verb-Verb Comparison
- Object-Object Comparison
- Subject-Verb Comparison
- Verb-Object Comparison
- Cross Subject-Object Comparison

For each of these comparisons, we assign a matching score of 1.0 (match) or 0.0 (mismatch).

2.3 Semantic Similarity Measures

2.3.1 Named Entity, Number, Time Expression Match and Mismatch

Careful observation of the development dataset revealed that mismatch of named entities, numbers or time expressions might cause semantic dissimilarity, for example, when s_1 consists of a named entity that does not appear in s_2. Based on this, we detect both match and mismatch of named entities, numbers and time expressions between each pair of sentences (similar to that of content words). We use the Stanford Named Entity Recognizer (Finkel et al., 2005) to detect named entities in sentences.

2.3.2 LDA-based measures

We build two Latent Dirichilet Allocation (LDA) models (Blei et al., 2003) from Wikipedia and the training dataset separately, using the Gensim (Řehůřek and Sojka, 2010) and Mallet (McCallum, 2002) software with 100 requested latent topics. Each sentence is represented by a vector using topics estimated by LDA. The similarity between two sentences is calculated as the cosine similarity between their corresponding vectors.

2.3.3 Word-representation-based measures

Word representation computes vector representations of each word based on its context from very large datasets, usually capturing both syntactic and semantic information of words. Given two sentences s_1 and s_2 (stop-words are removed), each word of the sentences is represented as a single vector. We develop two different strategies as follows:

Strategy 1 For each word w_i in s_1, we identify a word w_j most similar to w_i in s_2 by using cosine similarity measure. We define a measure $\mathscr{W}2\mathscr{V}(s_1, s_2)$ as follows:

$$\mathscr{W}2\mathscr{V}(s_1, s_2) = \frac{\sum\limits_{w_i \in s_1} \max\limits_{w_j \in s_2} cos(w_i, w_j)}{len(s_1)},$$

where $cos(w_i, w_j)$ is the cosine similarity between the word vectors of w_i and w_j. We also apply this strategy for each category of content words (noun, verb, adjective, and adverb) separately.

Strategy 2 We sum up all of the vectors of words that occur in each sentence and define a sentence similarity measure $\mathscr{S}2\mathscr{V}(s_1, s_2)$ as follows:

$$\mathscr{S}2\mathscr{V}(s_1, s_2) = cos(\sum_{w_i \in s_1} w_i, \sum_{w_j \in s_2} w_j),$$

For word representation, we use both the Word2vec model (Mikolov et al., 2013) trained on Google News and the GloVe model (Pennington et al., 2014) trained on Common Crawl data.

2.3.4 WordNet-based measures

WordNet (Miller, 1995) is a commonly used lexical database of English where words of the same meaning are grouped into synonym sets (synsets). By using information derived from WordNet, we construct some similarity measures as follows:

Strategy 1 This is similar to **Strategy 1** for word-representation-based measures, however, instead of using cosine similarity, we use the Wordnet path similarity (the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy).

Strategy 2 We determine some semantic relationships, e.g, synonym, antonym, and hypernym between sentences. The proportions of synonym word pairs, antonym word pairs, hypernym word pairs in two sentences (by normalizing with the harmonic mean of their number of content words) are taken as proxies of their semantic similarity.

3 System Description

3.1 Machine Learning Setup

The machine learning setup is described as follows:

Pre-processing The pre-processing phase includes tokenization, POS tagging, lemmatization, NER, syntactic parsing with the Stanford CoreNLP Toolkit (Manning et al., 2014). For some measures, we filter out punctuations and stop-words by using a pre-compiled stop-words list.

Feature Generation We run each of the similarity measures separately and use the resulting scores as features for a machine learning classifier. A feature is selected for our final model if it proves useful in improving the performance of the system.

Feature Combination The pre-computed similarity score vectors serve as features for this step. Our system utilizes a classifier combination approach, using a simple linear regression model to combine all the similarity measures. We use the trial dataset that comprises the 2012, 2013 and 2014

datasets to develop and train our model. In the development cycle, we used a training dataset consisting of 6842 sentence pairs and a test dataset consisting of 3750 sentence pairs, with gold standard scores. We use the WEKA machine learning toolkit (Hall et al., 2009) to perform our experiments.

Post-processing If the pre-processed sentences match, we set their similarity score to 5 regardless of the output of our classifier. If the classifier outputs an invalid similarity score s which is not in the score range [0-5], we set the similarity score to $f(s)$ defined as follows: $f(s) = \begin{cases} 0+\alpha & \texttt{if } s < 0 \\ 5-\alpha & \texttt{if } s > 5 \end{cases}$

In our experiments, the best value for α is 0.5.

3.2 Submitted System

TATO-1stWTW Because of our limited time, we submitted only one run to the SemEval-2015 Task 2a. After the development cycle, we identified about 30 main features out of a numerous number of features experimented. These features achieved the best performance on the training dataset. For our final system, we trained the classifier on a joint dataset of all known training datasets, instead of training a separate classifier for each individual dataset.

4 Results

4.1 Results on the 2014 Test Data

We evaluated our model on the 2014 test data comprising pairs of news headlines (headlines), pairs of glosses (OnWN), image descriptions (images), DEFT-related discussion forums (deft-forum) and news (deft-news), and tweet comments and newswire headline mappings (tweet-news). We used the 2012, 2013 datasets consisting of 6842 sentence pairs to train our model. The test dataset contains 3750 sentence pairs excluded from training. Our model was compared against the best performing system on the SemEval-2014 English STS sub-task (DLS@CU-run2) using the official scorer. The results are summarized in Table 1. With regard to Deft-forum and Tweets, our system outperformed the DLS@CU's system, we also achieved a higher score in the weighted mean across all datasets.

4.2 Results on the 2015 Test Data

The official score is based on the average of Pearson correlation. Besides Pearson correlations computed for individual datasets, including answers-forums, answers-students, belief, headlines, and images, Mean scores are provided to show the weighted means across all datasets (the weight is based on the number of sentence pairs in each dataset).

Run	DF	DN	H	I	OWN	TN	Mean
TATO1	.550	.748	.755	.807	.817	.777	.764
DLS@CU2	.483	.766	.765	.821	.859	.764	.761

Table 1: Results on the 2014 test datasets: deft-forum (DF), deft-news (DN), headlines (H), images (I), OnWN (OWN), tweet-news (TN).

Table 2 reports our official results achieved on the test data (**TATO-1stWTW**), besides the highest-performance and lowest-performance systems (according to Mean), and also the task baseline system. Our system was ranked among the most robust systems out of more than 70 participating systems and achieved good performance on answers-forums and belief datasets.

#	Run	AF	AS	B	H	I	Mean
1	DLS@CU1	.739	.773	.749	.825	.864	.802
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
25	TATO1	.680	.685	.721	.767	.817	.742
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	baseline1	.445	.665	.652	.531	.604	.587
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
73	DalGTM1	.290	-.053	.063	.060	.066	.062

Table 2: Official results on the test datasets: answers-forums (AF), answers-students (AS), belief (B), headlines (H), and images (I).

5 Conclusions and Future Work

This paper describes the TATO team's submission to the SemEval-2015 Task 2a: "Semantic Textual Similarity for English". Our system uses a simple linear regression model to combine multiple text similarity measures at different levels of depth: lexical, syntactic, and semantic. While we did not achieve the highest ranks on any of the particular datasets, our system was ranked among the most robust systems out of more than 70 participating systems.

For the future work, we will explore other evaluation measures for STS and try to train a separate classifier for each type of the existing datasets. We also suggest that we should work on some other types of data, such as legal or medical data.

References

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 435–440.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Johan Bos and Katja Markert. 2005. Recognising Textual Entailment with Logical Inference. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635.

Yee Seng Chan and Hwee Tou Ng. 2008. MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 55–62.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges*, Lecture Notes in Computer Science, pages 177–190.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 138–145.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

P. Foltz, W. Kintsch, and T. Landauer. 1998. The Measurement of Textual Coherence with Latent Semantic Analysis. In *Journal of the Discourse Processes*, 25(2&3):285–307.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1606–1611.

Wael H. Gomaa and Aly A. Fahmy. 2013. Article: A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13):13–18.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. An Introduction to Latent Semantic Analysis. In *Journal of the Discourse Processes*, 25:259–284.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. Tesla: Translation Evaluation of Sentences with Linear-programming-based Analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 354–359.

Prodromos Malakasiotis and Ion Androutsopoulos. 2007. Learning Textual Entailment Using SVMs and String Similarity measures. In *Proceedings of of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47.

Prodromos Malakasiotis. 2009. Paraphrase Recognition Using Machine Learning to Combine Similarity measures. In *Proceedings of the 47th Annual Meeting of ACL and the 4th Int. Joint Conference on Natural Language Processing of AFNLP*, pages 27–35.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1310.4546*.

George A. Miller. 1995. Wordnet: A Lexical Database for English. *Communications of the Association for Computing Machinery*, 38(11):39–41.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting Paraphrases in a Question Answering System. In *Proceedings of the 2nd International Workshop on Paraphrasing*, pages 25–32.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Matthew G. Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, 23(2-3):117–127.

Fabio massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2009. A Machine Learning Approach to Textual Entailment Recognition. *Natural Language Engineering*, 15(4):551–582.