MITRE at SemEval-2017 Task 1: Simple Semantic Similarity

John Henderson, Elizabeth M. Merkhofer, Laura Strickhart and **Guido Zarrella**
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420, USA
{jhndrsn,emerkhofer,lstrickhart,jzarrella}@mitre.org

Abstract

This paper describes MITRE's participation in the Semantic Textual Similarity task (SemEval-2017 Task 1), which evaluated machine learning approaches to the identification of similar meaning among text snippets in English, Arabic, Spanish, and Turkish. We detail the techniques we explored, ranging from simple bag-of-ngrams classifiers to neural architectures with varied attention and alignment mechanisms. Linear regression is used to tie the systems together into an ensemble submitted for evaluation. The resulting system is capable of matching human similarity ratings of image captions with correlations of 0.73 to 0.83 in monolingual settings and 0.68 to 0.78 in cross-lingual conditions.

1 Introduction

Semantic Textual Similarity (STS) measures the degree to which two snippets of text convey the same meaning. Cross-lingual STS measures the same for sentence pairs written in two different languages. Automatic identification of semantically similar text has practical applications in domains such as evaluation of machine translation outputs, discovery of parallel sentences in comparable corpora, essay grading, and news summarization. It serves as an easily explained assay for systems modeling semantics.

SemEval-2017 marked the sixth consecutive year of a shared task measuring progress in STS. Current machine learning approaches to measuring semantic similarity vary widely. One design decision for STS systems is whether to explicitly align words between paired sentences. Wieting et al. (2016) demonstrate that sentence embeddings without explicit alignment or attention can often provide reasonable performance on STS tasks. Related work in textual entailment offers evidence that neural models with soft alignment outperform embeddings-only approaches Chen et al. (2016); Parikh et al. (2016). However these results were obtained on a dataset multiple orders of magnitude larger than existing STS datasets. In absence of large datasets, word alignments similar to those used in statistical machine translation have proven to be useful (Zarrella et al., 2015; Itoh, 2016).

In this effort we explored diverse methods for aligning words in pairs of candidate sentences: translation-inspired hard word alignments as well as soft alignments learned by deep neural networks with attention. We also examined a variety of approaches for comparing aligned words, ranging from bag-of-ngrams features leveraging hand-engineered lexical databases, to recurrent and convolutional neural networks operating over distributed representations. Although an ideal cross-lingual STS system might operate directly on input sentences in their original language, we used machine translation to convert all the inputs into English. The paucity of in-domain training data and the simplicity of the image caption genre made the translation approach reasonable. Our contribution builds on approaches developed for English STS but points a way forward for progress on knowledge-lean, fully-supervised methods for semantic comparison across different languages.

2 Task, Data and Evaluation

Semantic Textual Similarity was a shared task organized within SemEval-2017 (Agirre et al., 2017). The task organizers released 1,750 sentence pairs of evaluation data organized into six tracks: Arabic, Spanish, and English monolingual, as well as Arabic-English, Spanish-English, and

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 185–190,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

Turkish-English cross-lingual.

Most of this evaluation data was sourced from the Stanford Natural Language Inference corpus (Bowman et al., 2015). The sentences are English-language image captions, grouped into pairs and human-annotated on a scale of 0 to 5 for semantic similarity. In the monolingual English task, the average sentence length was 8.7 words, and the average rating was 2.3 (e.g. *The woman had brown hair.* and *The woman has gray hair.*) There was a roughly balanced distribution of highly rated pairs (e.g. *A woman is bungee jumping.* and *A girl is bungee jumping.*) and poorly rated pairs (e.g. *The yard has a dog.* and *The dog is running after another dog.*) Annotated sentence pairs were manually translated from English into other languages to create additional tracks.

For each pair, task participants predicted a similarity score. Systems were evaluated by Pearson correlation with the human ratings.

3 System Overview

We created an ensemble of five systems which each independently predicted a similarity score. Some features were reused among many components, including word embeddings, machine translations, alignments, and dependency parses.

3.1 English Word Embeddings

We used word2vec (Mikolov et al., 2013) to learn distributed representations of words from the text of the English Wikipedia. We applied word2phrase twice to identify phrases of up to four words, and trained a skip-gram model of size 256 for the 630,902 vocabulary items which appeared at least 100 times, using a context window of 10 words and 15 negative samples per example.

3.2 Machine Translation

Sentences in the image caption genre tend to be short and use a simple vocabulary. To test the extent to which this is true of SNLI data, we trained a small unregularized neural language model which achieved a perplexity of 18.9 on a held-out test set. The same parameterization achieved a perplexity of 114.5 in experiments on the Penn Treebank (Zaremba et al., 2014). We proceeded to translate all non-English sentences to English, recognizing that modern MT systems are sufficient to provide high quality translations for simple sentences. We used the Google Translate API in mid-January 2017.

3.3 Dependency Parses

The dependency parse arcs were used as features to assist in aligning and comparing pairs of words. The Stanford Parser library produced these typed dependency representations (Chen and Manning, 2014). The English PCFG model with basic dependencies was used rather than the default collapsed dependencies to ensure that the parser gave us exactly one parse arc for each token.

3.4 Alignment

Comparing sentences can be a tallying process. One can find all associated atomic pairs in the left hand and right hand sides, cross them off, and judge the dissimilarity based on the remaining residuals. This process is reminiscent of finding translation equivalences for training machine translation systems (Al-Onaizan et al., 1999).

To this end, we built an alignment system on top of word embeddings. First, the *min alignment* is produced to maximize the sum of cosine similarities ($sim(w_i, w_j) = 1 + \cos(w_i, w_j)$) of word vectors corresponding to aligned word pairs under the constraint that no word is aligned more than once. The *max alignment* is constrained such that each word must be paired with at least one other, and the total number of edges in the alignment can be no more than word count of the longer string. In both cases, `LPSOLVE` was employed to find the assignment maximizing these criteria (Berkelaar et al., 2004).

Dependency parses constructed in Section 3.3 were aligned in a similar way. Consider dependency arcs $a_i : head \rightarrow dep$ Instead of the sum of cosine similarities as atoms in the linear program, however, we used $sim(a_1, a_2) = sim(head(a_1), head(a_2)) + 10sim(dep(a_1), dep(a_2))$ to give preference to matching dependency arcs a_1 and a_2 with similar heads.

3.5 Ensemble Components

TakeLab The open source TakeLab Semantic Text Similarity System was incorporated as a baseline (Šarić et al., 2012). Specifically we use LIBSVM to train a support vector regression model with an RBF kernel, cost parameter of 20, gamma of 0.2, and epsilon of 0.5. Input features were comprised of TakeLab-computed n-gram overlap and word similarity metrics.

Recurrent Convolutional Neural Network We recreate the recurrent neural network (RNN) model described in Zarrella et al. (2015) and train it using the embeddings and parse-aware alignments described above. Briefly, this 16-dimensional RNN operates over a sequence of aligned word pairs, comparing each pair according to features that encode embedding similarity, word position, and unsupervised string similarity.

We extended this model with four new feature categories. The first was a binary variable that indicates whether both words in the pair were determined to have the same dependency type in their respective parses. We also added three convolutional recurrent neural networks (CRNNs), each of which receive as input a sequence of word embeddings, and which learn STS features via 256 1D convolutional filters connected (with 50% dropout) to a 128-dimensional LSTM. For each aligned word pair, the first CRNN operates on the embeddings of the aligned words, the second CRNN operates on the squared difference of the embeddings of the aligned words, and the final CRNN operates on the embeddings of the parent words selected by the dependency parse. All above RNN outputs were concatenated to form a sequence of 400-dimensional (16+128*3) timesteps, which fed a 128-dimensional LSTM connected to a single sigmoidal output unit.

We unrolled this network to a zero-padded sequence length of 60 and trained it to convergence using Adam with a mean average error loss function (Kingma and Ba, 2014). The embeddings were not updated during training. We ensembled eight instances of this network trained from different random initializations.

Paris: String Similarity More than a decade ago, MITRE entered a system based on string similarity metrics in the 2004 Pascal RTE competition (Bayer et al., 2005). The `libparis` code base implements eight different string similarity and machine translation evaluation algorithms; measures include an implementation of the MT evaluation BLEU (Papineni et al., 2002); WER, a common speech recognition word error rate based on Levenshtein distance (Levenshtein, 1966); WER-g (Foster et al., 2003); ROUGE (Lin and Och, 2004); a simple position-independent error rate similar to PER (Leusch et al., 2003); both global and local similarity metrics often used for biological string comparison (Gusfield, 1997). Finally, there are precision and recall measures based on bags of *all* substrings (or n-grams in word tokenization).

In total, the package computes 22 metrics for a pair of strings. The metrics were run on both case-folded and original versions as well as on word tokens and characters, yielding 88 string similarity features. Some of the metrics are not symmetric, so they were run both forward and reversed based on presentation in the dataset yielding 176 features. Finally, for each feature value x, $\log(x)$ was added as a feature, producing a final count of 352 string similarity features. `LIBLINEAR` used these features to build a L1-regularized logistic regression model. This system was unchanged, except for retraining, from the system described in Zarrella et al. (2015)

Simple Alignment Measures Section 3.4 describes methods we used for aligning two strings. L2-regularized logistic regression was used to combine 16 simple features calculated as side-effects of alignment. Details are described in Zarrella et al. (2015).

Enhanced BiLSTM Inference Model (EBIM) We recreated the neural model described in Chen et al. (2016) which reports state-of-the-art performance on the task of finding entailment in the SNLI corpus. The model encodes each sentence with a bidirectional LSTM over word embeddings, uses a parameter-less attention mechanism to produce a soft alignment matrix for the two sentences, and then does inference over each timestep and its alignment using another LSTM. Two fully-connected layers complete the prediction. Chen et al. (2016) improves performance by concatenating the final LSTM representation from EBIM with that of a similar model where a modified LSTM operates over a syntax tree; we did not include this extension in our submission.

Our implementation kept most hyperparameters described in the paper. However, we used the word2vec embeddings described above and found that freezing the embeddings produced better performance for this small dataset. We also found our models worked better without dropout on the embedding layer. Where the original model chooses a class via softmax, we output a semantic similarity score trained to minimize mean squared error.

	Primary	Track 1 AR-AR	Track 2 AR-EN	Track 3 ES-ES	Track 4a ES-EN	Track 4b ES-EN news	Track 5 EN-EN	Track 6 TR-EN
Official Score	0.6590	0.7294	0.6753	0.8202	0.7802	0.1598	0.8053	0.6430
Corrected Score	**0.6687**	0.7294	0.6753	0.8202	0.7802	0.1598	**0.8329**	**0.6831**

Table 1: Pearson correlations on official test set. Corrected ensemble effects in bold.

	Factored		Ablated	
Component	dev	test	dev	test
TakeLab	.8724	.6503	.8739	.6454
CRNNs-8	.8621	.6379	.8846	.6551
Paris	.8074	.5524	.8891	.6666
EBIM	.7742	.4760	.8886	.6687
Align	.7607	.5037	**.8910**	**.6722**
All In			.8900	.6687

Table 2: Factored and ablated system components evaluated on our dev set and the official test set.

3.6 Ensemble

The semantic similarity estimates of the predictors described above contributed to the final prediction with a weighting determined by L2-regularized logistic regression.

4 Experiment Details

We used as training data a selection of English monolingual sentence pairs released during prior SemEval STS evaluations. Specifically, we trained on 6,898 pairs of news and caption genre data from the 2012-2014 and 2016 evaluations. We used an additional 400 and 350 captions from the 2015 evaluation as development and tuning sets, respectively. We did not use out-of-genre data (e.g. dictionary definitions, Europarl, web forums, student essays) or the newly-released multilingual 2017 training data. The dev set was used to select hyperparameters for individual components, while the tuning set was used to select the hyperparameters for the final ensemble.

5 Results

The evaluation of our components on the competition test set is shown in Table 1. The official similarity score produced by this approach achieved 0.6590 correlation with expert judgment averaged across all tracks. A misfiling during construction of the ensemble submission for tracks 5 and 6 reduced the official score from 0.6687.

The dev columns of Table 2 show the ability of each individual system in isolation on the dev data (”Factored”) as well as the performance of the ensemble when the individual system was removed (”Ablated”). Note that the *Align* system should have been ablated from the final system to achieve a higher score. Presumably its capability was strictly dominated by the CRNNs that used many of the same features.

The test scores for individual CRNN models ranged from 0.605 to 0.636, highlighting the volatility inherent in the process. The CRNN-ensemble improved slightly over the best single model, with a score of 0.638.

6 Conclusion

Five models of semantic similarity constructed from 2004 to 2016 were combined for paraphrase detection in image captions. The TakeLab bag-of-features SVM developed and open-sourced in 2012, when trained on our selection of in-genre data and evaluated on a machine translated version of the test set, performed well enough in isolation to place fourth out of seventeen in the Primary Track of the Semantic Textual Similarity competition organized within SemEval-2017 Task 1, which had submissions from 31 teams in total.

Inclusion of explicit word alignments, a neural attention model, and recurrent networks accounting for sequences of syntactic dependencies yielded an improvement in Pearson correlation from 0.650 to 0.672, a modest improvement which increased the corrected system’s ranking to third. This surprising result is perhaps an indication that image captions have few of the complex linguistic dependencies that typically make estimating semantic similarity a difficult task. Future work could focus on testing whether this result holds when performing crosslingual STS without explicit machine translation.

Acknowledgments

This work was funded under the MITRE Innovation Program. Approved for Public Release; Distribution Unlimited: 17-0970.

References

Eneko Agirre, Daniel Cer, Mona Diab, Iigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*.

Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, F-J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. 1999. Statistical machine translation: Final report. Technical report, JHU Center for Language and Speech Processing.

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. 2005. MITRE's submissions to the EU Pascal RTE challenge. In *Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*.

Michel Berkelaar, Kjell Eikland, and Peter Notebaert. 2004. lp_solve 5.5, open source (mixed-integer) linear programming system. Software. http://lpsolve.sourceforge.net/5.5/.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. http://www.anthology.aclweb.org/D/D15/D15-1075.pdf.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*. http://www.aclweb.org/anthology/D14-1082.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *arXiv preprint arXiv:1609.06038* .

George Foster, Simona Gandrabur, Cyril Goutte, Erin Fitzgerald, Alberto Sanchis, Nicola Ueffing, John Blatz, and Alex Kulesza. 2003. Confidence estimation for machine translation. Technical report, JHU Center for Language and Speech Processing.

Dan Gusfield. 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Hideo Itoh. 2016. RICOH at SemEval-2016 task 1: IR-based semantic textual similarity estimation. *Proceedings of SemEval* https://www.aclweb.org/anthology/S/S16/S16-1106.pdf.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

G. Leusch, N. Ueffing, and H. Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proc. of the Ninth MT Summit*.

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10(8):707–710.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*. Geneva, Switzerland. http://www.aclweb.org/anthology/C04-1072.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

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*. ACL '02.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* .

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. http://www.aclweb.org/anthology/S12-1060.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceedings of ICLR*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .

Guido Zarrella, John Henderson, Elizabeth M Merkhofer, and Laura Strickhart. 2015. MITRE: Seven systems for semantic similarity in tweets. *Proceedings of SemEval* http://www.aclweb.org/anthology/S15-2002.

A Supplemental Material

These *min alignment* examples all come from Track 5.

Example 1: Similarity 5.0.

	the	boy	is	taking	a	test	at	school
a	1.69	1.32	1.56	1.36	$\boxed{2.00}$	1.26	1.47	1.31
boy	1.30	$\boxed{2.00}$	1.22	1.30	1.32	1.21	1.23	1.41
is	1.58	1.22	$\boxed{2.00}$	1.28	1.56	1.24	1.48	1.30
at	1.56	1.23	1.48	1.35	1.47	1.25	$\boxed{2.00}$	1.34
school	1.28	1.41	1.30	1.21	1.31	1.24	1.34	$\boxed{2.00}$
taking	1.39	1.30	1.28	$\boxed{2.00}$	1.36	1.34	1.35	1.21
a	$\boxed{1.69}$	1.32	1.56	1.36	2.00	1.26	1.47	1.31
test	1.23	1.21	1.24	1.34	1.26	$\boxed{2.00}$	1.25	1.24

Example 2: Similarity 2.6.

	two	men	standing	in	the	surf	on	a	beach
a	1.41	1.29	1.36	1.64	1.69	1.20	1.52	$\boxed{2.00}$	1.28
pair	$\boxed{1.54}$	1.33	1.38	1.35	1.40	1.21	1.34	1.37	1.28
of	1.42	1.34	1.35	$\boxed{1.66}$	1.79	1.18	1.53	1.60	1.30
men	1.29	$\boxed{2.00}$	1.40	1.35	1.36	1.25	1.27	1.29	1.33
walk_along	1.26	1.25	1.43	1.25	1.30	$\boxed{1.44}$	1.39	1.29	1.60
the	1.47	1.36	1.42	1.73	$\boxed{2.00}$	1.25	1.57	1.69	1.30
beach	1.31	1.33	1.36	1.30	1.30	1.66	1.33	1.28	$\boxed{2.00}$

Example 3: Similarity 0.0.

	men	are	trying	to	remove	oil	from	a_body	of	water
adding	1.12	1.21	1.29	1.21	$\boxed{1.40}$	1.07	1.18	1.15	1.16	1.18
aspirin	1.16	$\boxed{1.15}$	1.17	1.17	1.23	1.32	1.15	1.11	1.17	1.27
to	1.33	1.44	1.31	$\boxed{2.00}$	1.34	1.25	1.62	1.06	1.59	1.35
the	1.36	1.49	1.26	1.64	1.23	1.31	$\boxed{1.58}$	1.10	1.79	1.37
water	1.21	1.32	1.10	1.35	1.21	1.50	1.34	1.10	1.36	$\boxed{2.00}$
could	1.31	1.34	$\boxed{1.51}$	1.48	1.36	1.20	1.30	1.13	1.31	1.18
kill	$\boxed{1.30}$	1.26	1.41	1.35	1.44	1.18	1.27	1.19	1.32	1.19
the	1.36	1.49	1.26	1.64	1.23	1.31	1.58	1.10	$\boxed{1.79}$	1.37
plant	1.19	1.30	1.18	1.33	1.26	$\boxed{1.47}$	1.29	1.07	1.32	1.41