HENRY-CORE: Domain Adaptation and Stacking for Text Similarity*

Michael Heilman and **Nitin Madnani**
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541, USA
{mheilman,nmadnani}@ets.org

Abstract

This paper describes a system for automatically measuring the semantic similarity between two texts, which was the aim of the 2013 Semantic Textual Similarity (STS) task (Agirre et al., 2013). For the 2012 STS task, Heilman and Madnani (2012) submitted the PERP system, which performed competitively in relation to other submissions. However, approaches including word and n-gram features also performed well (Bär et al., 2012; Šarić et al., 2012), and the 2013 STS task focused more on predicting similarity for text pairs from new domains. Therefore, for the three variations of our system that we were allowed to submit, we used stacking (Wolpert, 1992) to combine PERP with word and n-gram features and applied the domain adaptation approach outlined by Daume III (2007) to facilitate generalization to new domains. Our submissions performed well at most subtasks, particularly at measuring the similarity of news headlines, where one of our submissions ranked 2nd among 89 from 34 teams, but there is still room for improvement.

1 Introduction

We aim to develop an automatic measure of the semantic similarity between two short texts (e.g., sentences). Such a measure could be useful for various applications, including automated short answer scoring (Leacock and Chodorow, 2003; Nielsen et al., 2008), question answering (Wang et al., 2007), and machine translation evaluation (Przybocki et al., 2009).

In this paper, we describe our submissions to the 2013 Semantic Textual Similarity (STS) task (Agirre et al., 2013), which evaluated implementations of text-to-text similarity measures. Submissions were evaluated according to Pearson correlations between gold standard similarity values acquired from human raters and machine-produced similarity values. Teams were allowed to submit up to three submissions. For each submission, correlations were calculated separately for four subtasks: measuring similarity between news headlines ("headlines"), between machine translation outputs and human reference translations ("SMT"), between word glosses from OntoNotes (Pradhan and Xue, 2009) and WordNet (Fellbaum, 1998) ("OnWN"), and between frame descriptions from FrameNet (Fillmore et al., 2003) and glosses from WordNet ("FNWN"). A weighted mean of the correlations was also computed as an overall evaluation metric (the OnWn and FNWN datasets were smaller than the headlines and SMT datasets).

The suggested training data for the 2013 STS task was the data from the 2012 STS task (Agirre et al., 2012), including both the training and test sets for that year. The 2012 task was similar except that the data were from a different set of subtasks: measuring similarity between sentences from the Microsoft Research Paraphrase corpus (Dolan et al., 2004) ("MSRpar"), between sentences from the Microsoft Research Video Description corpus (Chen and Dolan, 2011) ("MSRvid"), and between human and machine translations of parliamentary

[*] System description papers for this task were required to have a team ID and task ID (e.g., "HENRY-CORE") as a prefix.

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 96–102, Atlanta, Georgia, June 13-14, 2013. ©2013 Association for Computational Linguistics

proceedings (“SMTeuroparl”). The 2012 task provided training and test sets for those three subtasks and also included two additional tasks with just test sets: a similar OnWN task, and measuring similarity between human and machine translations of news broadcasts (“SMTnews”).

Heilman and Madnani (2012) described the PERP system and submitted it to the 2012 STS task. PERP measures the similarity of a sentence pair by finding a sequence of edit operations (e.g., insertions, deletions, substitutions, and shifts) that converts one sentence to the other. It then uses various features of the edits, with weights learned from labeled sentence pairs, to assign a similarity score. PERP performed well, ranking 7th out of 88 submissions from 35 teams according to the weighted mean correlation. However, PERP lacked some of the useful word and n-gram overlap features included in some of the other top-performing submissions. In addition, domain adaptation seemed more relevant for the STS 2013 task since in-domain data was available only for one (OnWN) of the four subtasks.

Therefore, in this work, we combine the PERP system with various word and n-gram features. We also apply the domain adaptation technique of Daume III (2007) to support generalization beyond the domains in the training data.

2 System Details

In this section, we describe the system we developed, and the variations of it that comprise our submissions to the 2013 STS task.

Our system is a linear model estimated using ridge regression, as implemented in the scikit-learn toolkit (Pedregosa et al., 2011). The system uses a 5-fold cross-validation grid search to tune the α penalty for ridge regression (with $\alpha \in 2^{\{-5,-4,\ldots,4\}}$). During development, we evaluated its performance on the full STS 2012 data (training and test) using 10-fold cross-validation, with the 5-fold cross-validation being used to tune within each training partition.

2.1 Features

Our full system uses the following features computed from an input sentence pair (s_1, s_2).

The system standardizes feature values to zero mean and unit variance by subtracting the feature’s mean and dividing by its standard deviation. The means and standard deviations are estimated from the training set, or from each training partition during cross-validation.

2.1.1 n-gram Overlap Features

The system computes Jaccard similarity (i.e., the ratio of the sizes of the set intersection to the set union) for the following overlap features:

- character n-gram overlap ($n = 1 \ldots 12$). Note that this is computed from the entire original texts for a pair, including punctuation, whitespace, etc.
- word n-gram overlap ($n = 2 \ldots 8$). We do not include $n = 1$ here because it would be identical to the $n = 1$ version for the unordered word n-gram feature described next.
- unordered word n-gram overlap features ($n = 1 \ldots 3$). By unordered, we mean combinations (in the mathematical sense of “combinations”) of word tokens, regardless of order. Note that these features are similar to the word n-gram overlap features except that the words need not be contiguous to match. For example, the text “John saw Mary” would result in the following unordered word n-grams: {*john*}, {*mary*}, {*saw*}, {*john, saw*}, {*mary, saw*}, {*john, mary*}, and {*john, mary, saw*}.

For the word and unordered n-gram overlap features, we computed two variants: one based on all tokens and one based on just content words, which we define as words that are not punctuation and do not appear in the NLTK (Bird et al., 2009) English stopword list. We lowercase everything for the word overlap measures but not for character overlap.

2.1.2 Length Features

The system includes various length-related features, where $\mathrm{L}_{max} = \max(\mathrm{length}(s_1), \mathrm{length}(s_2))$, $\mathrm{L}_{min} = \min(\mathrm{length}(s_1), \mathrm{length}(s_2))$, and $\mathrm{length}(x)$ denotes the number of tokens in x. $\log$ denotes the natural logarithm.

- $\log(\frac{\mathrm{L}_{max}}{\mathrm{L}_{min}})$
- $\frac{\mathrm{L}_{max}-\mathrm{L}_{min}}{\mathrm{L}_{max}}$

- $\log(\mathrm{L}_{min})$
- $\log(\mathrm{L}_{max})$
- $\log(|\mathrm{L}_{max} - \mathrm{L}_{min}| + 1)$

2.1.3 Sentiment Features

The system includes various features based on the proprietary sentiment lexicon described by Beigman Klebanov et al. (2012). Each word in this lexicon is associated with a 3-tuple specifying a distribution over three classes: positive, negative, and neutral. These distributions were estimated via crowdsourcing. If a word is not in the lexicon, we assume its positivity and negativity are zero.

We define the set of sentiment words in a sentence s as $\sigma(s) = \{w : \text{positivity}(w) > 0.5 \vee \text{negativity}(w) > 0.5\}$. We also define the positivity, negativity, and neutrality of a sentence as the sum over the corresponding values of individual words w. For example, $\text{positivity}(s) = \sum_{w \in s} \text{positivity}(w)$.

The system includes the following features:

- $\frac{\sigma(s_1) \cap \sigma(s_2)}{\sigma(s_1) \cup \sigma(s_2)}$ (i.e., the Jaccard similarity of the sentiment words)
- The cosine distance between $(\text{positivity}(s_1), \text{negativity}(s_1))$ and $(\text{positivity}(s_2), \text{negativity}(s_2))$
- $|\text{positivity}(s_1) - \text{positivity}(s_2)|$
- $|\text{negativity}(s_1) - \text{negativity}(s_2)|$
- $|\text{neutrality}(s_1) - \text{neutrality}(s_2)|$

2.1.4 PERP with Stacking

The system also incorporates the PERP system (Heilman and Madnani, 2012) (as briefly described in §1) as a feature in its model by using 10-fold stacking (Wolpert, 1992). Stacking is a procedure similar to k-fold cross-validation that allows one to use the output of one model as the input to another model, without requiring multiple training sets. A PERP model is iteratively trained on nine folds and then the PERP feature is computed for the tenth, producing PERP features for the whole training set, which are then used in the final regression model.

We trained PERP in a general manner using data from all the STS 2012 subtasks rather than training subtask-specific models. PERP was trained for 100 iterations.

We refer readers to Heilman and Madnani (2012) for a full description of PERP. Next, we provide details about modifications made to PERP since STS 2012. Although these details are not necessary to understand how the system works in general, we include them here for completeness.

- We extended PERP to model abbreviations as zero cost edits, using a list of common abbreviations extracted from Wikipedia.[1]
- In a similar vein, we also extended PERP to model multiword sequences with differing punctuation (e.g., “Built-In Test” → “Built In Test”) as zero cost edits.
- We changed the stemming and synonymy edits in the original PERP (Heilman and Madnani, 2012) to be substitution edits that activate additional stemming and synonymy indicator features.
- We added an incentive to TERp’s (Snover et al., 2009) original inference algorithm to prefer matching words when searching for a good edit sequence. We added this to avoid rare cases where other edits would have a negative costs, and then the same word in a sentence pair would be, for example inserted and deleted rather than matched.
- We fixed a minor bug in the inference algorithm, which appeared to only affect results on the MSRvid subtask in the STS 2012 task.
- We tweaked the learning algorithm by increasing the learning rate and not performing weight averaging.

2.2 Domain Adaptation

The system also uses the domain adaptation technique described by Daume III (2007) to facilitate generalization to new domains. Instead of having a single weight for each of the features described above, the system maintains a generic and a subtask-specific copy. For example, the content bigram overlap feature had six copies: a generic copy and one for each of the five subtasks in the training data from

[1] http://en.wikipedia.org/wiki/List_of_acronyms_and_initialisms, downloaded April 27, 2012

STS 2012 (i.e., OnWN, MSRpar, MSRvid, SMTeuroparl, SMTnews). And then for an instance from MSRpar, only the generic and MSRpar-specific versions of the feature will be active. For an instance from a new subtask (e.g., a test set instance), only the generic feature will be active.

We also included a generic intercept feature and intercept features for each subtask (these always had a value of 1). These help the model capture, for example, whether high or low similarities are more frequent in general, without having to use the other feature weights to do so.

2.3 Submissions

We submitted three variations of the system.

- **Run 1**: This run used all the features described above. In addition, we mapped the test subtasks to the training subtasks as follows so that the specific features would be active for test data from previously unseen but related subtasks: headlines to MSRpar, SMT to SMTnews, and FNWN to OnWN.
- **Run 2**: As in Run 1, this run used all the features described above. However, we did not map the STS 2013 subtasks to STS 2012 subtasks. Thus, the specific copies of features were only active for OnWN test set examples.
- **Run 3**: This run used all the features except for the PERP and sentiment features. Like Run 2, this run did not map subtasks.

3 Results

This section presents results on the STS 2012 data (our development set) and results for our submissions to STS 2013.

3.1 STS 2012 (development set)

Although we used cross-validation on the entire STS 2012 dataset during preliminary experiments (§2), in this section, we train the system on the original STS 2012 training set and report performance on the original STS 2012 test set, in order to facilitate comparison to submissions to that task. It is important to note that our system's results here may be somewhat optimistic since we had access to the STS 2012 test data and were using it for development, whereas the participants in the 2012 task only had access to the training data.

Table 1 presents the results. We include the results for our three submissions, the results for the top-ranked submission according to the weighted mean ("UKP"), the results for the best submission from Heilman and Madnani (2012) ("PERPphrases"), and the mean across all submissions. Note that while we compare to the PERP submission from Heilman and Madnani (2012), the results are not directly comparable since the version of PERP is not the same and since PERP was trained differently.

For Run 1 on the STS 2012 data, we mapped OnWN to MSRpar, and SMTnews to SMTeuroparl, similar to Heilman and Madnani (2012).

3.2 STS 2013 (unseen test set)

Table 2 presents results for our submissions to the 2013 STS task. We include results for our three submissions, results for the top-ranked submission according to the weighted mean, results for the baseline provided by the task organizers, and the mean across all submissions and the baseline from the organizers.[2]

Note that while our Run 2 submission outperformed the top-ranked UMBC submission on the headlines subtask, as shown in 2, there was another UMBC submission that performed better than Run 2 for the headlines subtask.

4 Discussion

The weighted mean correlation across tasks for our submissions was relatively poor compared to the top-ranked systems for STS 2013: our Run 1, Run 2, and Run 3 submissions beat the baseline and ranked 41st, 26th, and 48th, respectively, out of 89 submissions.

The primary reason for this result is that performance of our submissions was poor for the OnWN subtask, where, e.g., our Run 2 submission's correlation was $r = .4631$, compared to $r = .8431$ for the top-ranked submission for that subtask ("deft-baseline"). Upon investigation, we found that OnWN training and test data were very different in terms of their score distributions. The mean gold

[2] The STS 2013 results are from `http://ixa2.si.ehu.es/sts/`.

Submission	MSRpar	MSRvid	SMTeuroparl	OnWN	SMTnews	W. Mean
Run 1	.6461	.8060	.5014	.7073	.4876	.6577
Run 2	.6461	.8060	.5014	.7274	.4744	.6609
Run 3	.6369	.7904	.5101	.7010	.4985	.6529
UKP (top-ranked)	.6830	.8739	.5280	.6641	.4937	.6773
PERPphrases	.6397	.7200	.4850	.7124	.5312	.6399
mean-2012	.4894	.7049	.3958	.5557	.3731	.5286

Table 1: Pearson correlations for STS 2012 data for each subtask and then the weighted mean across subtasks. "UKP" was submitted by Bär et al. (2012), "PERPphrases" was submitted by Heilman and Madnani (2012), and "mean-2012" is the mean of all submissions to STS 2012.

Submission	headlines	OnWN	FNWN	SMT	W. Mean
Run 1	.7601	.4631	.3516	.2801	.4917
Run 2	.7645	.4631	.3905	.3593	.5229
Run 3	.7103	.3934	.3364	.3308	.4734
UMBC (top-ranked)	.7642	.7529	.5818	.3804	.6181
baseline	.5399	.2828	.2146	.2861	.3639
mean-2013	.6022	.5042	.2887	.2989	.4503

Table 2: Pearson correlations for STS 2013 data for each subtask and then the weighted mean across subtasks. "UMBC" = "UMBC_EBIQUITY-ParingWords", and "mean-2013" is the mean of the submissions to STS 2013 and the baseline.

standard similarity value for the STS 2012 OnWN data was 3.87 (with a standard deviation of 1.02), while the mean for the 2013 OnWN data was 2.31 (with a standard deviation of 1.76). We speculate that our system performed relatively poorly because it was expecting the OnWN data to include many highly similar sentences (as in the 2012 data). We hypothesize that incorporating more detailed WordNet information (only the PERP feature used WordNet, and only in a limited fashion, to check synonymy) and task-specific features for comparing definitions might have helped performance for the OnWN subtask.

If we ignore the definition comparison subtasks, and consider performance on just the headlines and SMT subtasks, the system performed quite well. Our Run 2 submission had a mean correlation of $r = .5619$ for those two subtasks, which would rank 5th among all submissions.

We have not fully explored the effects on performance of the domain adaptation approach used in the system, but our approach of mapping tasks used for our Run 1 submission did not seem to help. It seems better to keep a general model, as in Runs 2 and 3.

Additionally, we observe that the performance of Run 3, which did not use the PERP and sentiment features, was relatively good compared to Runs 1 and 2, which used all the features. This indicates that if speed and implementation simplicity are important concerns for an application, it may suffice to use relatively simple overlap and length features to measure semantic similarity.

The contribution of domain adaptation is not clear. Mapping novel subtasks to tasks for which training data is available (§2.3), in combination with the domain adaptation technique we used, did not generally improve performance. However, we leave to future work a detailed analysis of whether the domain adaptation approach (without mapping) is better than simply training a separate system for each subtask and using out-of-domain data when in-domain data is unavailable.

5 Conclusion

In this paper, we described a system for predicting the semantic similarity of two short texts. The system uses stacking to combine a trained edit-based similarity model (Heilman and Madnani, 2012) with simple features such as word and n-gram overlap, and it uses the technique described by Daume III (2007) to support generalization to domains not represented in the training data. We also presented evaluation results, using data from the STS 2012 and STS 2013 shared tasks, that indicate that the system performs competitively relative to other approaches for many tasks. In particular, we observed very good performance on the news headline similarity and MT evaluation subtasks of the STS 2013 shared task.

Acknowledgments

We would like to thank the STS 2013 task organizers for facilitating this research and Dan Blanchard for helping with scikit-learn.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

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.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. UKP: Computing semantic textual similarity by combining multiple content similarity measures. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 435–440, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Beata Beigman Klebanov, Jill Burstein, Nitin Madnani, Adam Faulkner, and Joel Tetreault. 2012. Building sentiment lexicon(s) from scratch for essay data. In *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, New Delhi, India, March.

S. Bird, E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June. Association for Computational Linguistics.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Coling 2004*, pages 350–356, Geneva, Switzerland, Aug 23–Aug 27. COLING.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250.

Michael Heilman and Nitin Madnani. 2012. ETS: Discriminative edit models for paraphrase scoring. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 529–535, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

C. Leacock and M. Chodorow. 2003. c-rater: Scoring of short-answer questions. *Computers and the Humanities*, 37.

Rodney D. Nielsen, Wayne Ward, and James H. Martin. 2008. Classification errors in a domain-independent assessment system. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18, Columbus, Ohio, June. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

S. S. Pradhan and N. Xue. 2009. OntoNotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North*

American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts, pages 11–12.

M. A. Przybocki, K. Peterson, S. Bronsart, and G. A. Sanders. 2009. The NIST 2008 metrics for machine translation challenge - overview, methodology, metrics, and results. *Machine Translation*, 23(2-3):71–103.

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, September.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for measuring semantic text similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic, June. Association for Computational Linguistics.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.