MITRE: Seven Systems for Semantic Similarity in Tweets

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

Abstract

This paper describes MITRE's participation in the Paraphrase and Semantic Similarity in Twitter task (SemEval-2015 Task 1). This effort placed first in Semantic Similarity and second in Paraphrase Identification with scores of Pearson's r of 61.9%, F1 of 66.7%, and maxF1 of 72.4%. We detail the approaches we explored including mixtures of string matching metrics, alignments using tweet-specific distributed word representations, recurrent neural networks for modeling similarity with those alignments, and distance measurements on pooled latent semantic features. Logistic regression is used to tie the systems together into the ensembles submitted for evaluation.

1 Introduction

Paraphrase identification is the task of judging if two texts express the same or very similar meaning. Automatic identification of paraphrases has practical applications for a range of domains, including news summarization, information retrieval, essay grading, and evaluation of machine translation outputs. Furthermore, work on paraphrase detection tends to advance the state of art in modeling semantics and semantic similarity in natural language in general.

Current approaches to paraphrase detection vary widely. The Microsoft Research Paraphrase Corpus, with pairs of sentences from newswire text, serves as a benchmark for the task (Dolan et al., 2004). One top result on this dataset uses features from surface characteristics of text (Madnani et al., 2012). Another system with comparable results models sentences as hierarchical compositions of distributed word embeddings (Socher et al., 2011). SemEval-2015 Task 1 (Xu et al., 2015), with a corpus drawn from Twitter, offers an opportunity to test paraphrase systems in a domain with an expanded vocabulary and informal grammar.

Our contribution builds upon the recent success of distributed representations of language (Mikolov et al., 2013a; Pennington et al., 2014). We further aim to minimize reliance on language- and domain-dependent tools. However we do not possess enough labeled paraphrase data to train a generalized model of word composition. Instead we explore models that examine low-dimensional relationships between individual pairs of aligned words, and combine the above with string similarity features that generalize well to out-of-vocabulary terms.

In the remainder of this paper, we describe our high-performing system for modeling semantic similarity between two tweets. In Section 2 we describe the data, task, and evaluation. In Section 3 we discuss details of systems we built to solve the semantic similarity task. We describe our experiments on different parameterizations in Section 4. In Section 5 we present performance results for our ensembles and all subsystems, and in Section 6 we summarize our findings.

2 Task, data and evaluation

Paraphrase and Semantic Similarity in Twitter was a shared task organized within SemEval-2015.

The task organizers released 18,762 pairs of English-language tweets with a 70/25/5 split for train, development, and test sets. The organizers removed URLs, deleted non-alphanumeric characters, and provided part of speech tags. Tweet pairs were judged by five human annotators to be a paraphrase (e.g. *Amber alert gave me a damn heart attack* and *That Amber alert scared the crap out of me*) or not (e.g. *My phone is annoying me with these amber alert* and *Am I the only one who dont get Amber alert*). Approximately 35% of provided pairs are paraphrases. For each pair, task participants predict

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 12–17,
Denver, Colorado, June 4-5, 2015.

a binary label and optionally provide a confidence score. Systems were evaluated by F1 measure, F1 at the best confidence threshold, and Pearson correlation with expert annotation.

3 System overview

We created an ensemble of seven systems which each independently predicted a semantic similarity score. Some features were reused among the components, including word embeddings and alignments.

3.1 Twitter Word Embeddings

We used word2vec to learn distributed representations of words and phrases from an unlabeled corpus of 330.3 million tweets sampled in 2013 from Twitter's public streaming API. Retweets and non-English messages were not included in the sample. Text was lowercased and processed to mimic the style of the task data. We applied word2phrase (Mikolov et al., 2013b) twice consecutively to identify phrases comprised of up to four words. We then trained a skip-gram model of size 256 for the 1.87 million vocabulary items which appeared at least 25 times, using a context window of 10 words and 15 negative samples per positive example. These hyperparameters were selected based on our prior experience in training embeddings for identification of word analogies.

3.2 Alignment

Comparing semantics in two tweets can be imagined as a tallying process. One finds some semantic atom on the left hand side and searches for it in the right hand side. If found, it gets crossed off. Otherwise, that atom contributes to a difference. Repeat on the other side. This idealized 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. Each tweet was converted into a bag of words, and two different alignments were created. The *min alignment* maximized the cosine similarity of aligned pairs under the constraint that no word could be aligned more than once. The *max alignment* was 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. `LPSOLVE` was employed to find the assignment maximizing these criteria (Berkelaar et al., 2004).

3.3 Seven Systems

Random Projection The random projection family of Locality Sensitive Hashing algorithms is a probabilistic technique for reducing high dimensional inputs to a fixed-length low dimensional sketch (Charikar, 2002), in which similar inputs yield similar hashes. This characteristic is useful for approximate nearest neighbor search and online clustering (Petrović et al., 2010), but we use it here to obtain an unsupervised similarity metric that identifies string overlap at many levels of granularity. Concretely, we extract the set of all word unigrams, word bigrams, and character n-grams of lengths 2 through 5. These features are input to 2048 independent binary classifiers with random weights, and each classifier contributes a single bit to the resulting hash. We assess similarity of two tweets by measuring the Hamming distance between their bit vectors.

Recurrent Neural Network One common approach to paraphrase detection is to construct a model of each sentence before learning a distance function over these representations. We chose to sidestep this global semantics modeling problem and instead directly measured the relationships between embedded lexical items.

In particular, we used a Recurrent Neural Network to examine the sequence of aligned word pairs obtained from the min alignment process described in section 3.2. For each aligned pair, we computed descriptive statistics that were used as input to the network: cosine similarity and Euclidean distance of the aligned word embeddings, the magnitudes of each word's vector, and the relative position of each word in the sentence. These features enabled the network to consider the quality of the alignment without introducing sparsity by including the word vectors themselves. The RNN also received two global features at each time step: the ratio of sentence lengths and the normalized Hamming distance computed via random projection as described above.

The RNN contained 8 input features, 16 hidden units, and a single output, composed as an Elman network (Elman, 1990) with tied weights.

We unfolded it using backpropagation through time (Williams and Zipser, 1990) to create a deep network with as many hidden layers as there were lexical units in the shorter sentence. We trained the RNN with stochastic gradient descent and a formulation of dropout (Hinton et al., 2012) that randomly removed a single word pair from each training sequence. Parameters were tuned on the development set, including a minibatch of 20, a learning rate of 0.05 or 0.06, hyperbolic tangent activation functions, and early stopping after about 2000 iterations. Two RNNs were used in the final ensemble, each trained with different learning rates.

Paris: String Similarity MITRE entered a system based on string similarity metrics in the 2004 Pascal RTE competition (Bayer et al., 2005). We revivified the code base (called `libparis`) and updated it for this evaluation. Eight different string similarity and machine translation evaluation approaches are implemented in this package; 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, an error rate similar to WER, but with denominator based on the min edit traceback (Foster et al., 2003); the MT evaluation ROUGE (Lin and Och, 2004); a simple position-independent error rate similar to PER as described in Leusch et al. (2003); both global and local similarity metrics often used for biological string comparison as described in Gusfield (1997). Finally, there are precision and recall measures based on bags of *all* substrings (or n-grams in word tokenization).

In total we computed 22 metrics for a pair of strings. The metrics were run on both lowercased 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. We used `LIBLINEAR` with these features to build a L1-regularized logistic regression model.

Simple Alignment Measures Section 3.2 describes methods we used for aligning two strings. We built one component that computed similarity between tweets using simple metrics applied only to the aligned word pairs. Mean vectors and pooled component-wise min and max vectors were computed for both sides of the two different types of alignments. Those six pairs of vectors were compared using cosine distance, Manhattan distance, and Euclidean distance, resulting in eighteen features. Separately, the alignments were traversed and pairs of word vectors were compared using the three distance functions. The means of those comparisons produced six more features. L2-regularized logistic regression combined these 24 features into a single measure of semantic similarity.

Similarity Matrices, Averaged and Min/Max Two subsystems drew upon a similarity matrix and dynamic pooling technique presented in Socher et al. (2011). This method considers distance between all syntactically meaningful subunits of two sentences. First, a representation is induced for each node of the parse tree of two sentences, starting from word embeddings at leaf nodes. Then a similarity matrix is created from measurements of Euclidean distance between every pair of nodes. Finally, a dynamic pooling scheme reduces this to a fixed-size representation that is used as input to a logistic regression classifier. For one subsystem in MITRE's contribution, nodes were represented as averages of their child nodes; for another, nodes were represented as the concatenation of the minimum and maximum of the child nodes.

Normalized Averages This subsystem computed an unsupervised distance metric based on semantic features. We first replaced each word in the tweet with its synonym from the Twitter normalization lexicon (Han and Baldwin, 2011), for example converting *tv* to *television*. The embeddings of these words were used in experiments on weighted averaging and pooling, folding of part-of-speech tags, and various distance and similarity metrics. The best F1 score on the development set was achieved by averaging the word vectors and computing Euclidean distance between the two tweets' resulting vectors.

3.4 Ensembles

The predictors described above were selected for inclusion in a larger ensemble on the basis of their

Name	Factored	Ablated
BLEU	61.5	64.6
ROUGE	60.2	63.8
PER	60.0	64.4
substring bags	58.7	63.5
WER	58.0	63.9
WER-g	57.9	63.9
global sim	57.7	64.1
local sim	55.9	63.1
none	–	63.9

Table 1: Dev set F1 scores for string similarities.

performance on the development set. Each component's semantic similarity score contributed to the final prediction with a weighting determined by L2-regularized logistic regression. Binary paraphrase labels were assigned by choosing an ensemble score threshold that optimized development set F1.

The ensemble described in this paper was submitted for scoring under the name MITRE IKR. A second submission was identical with one exception: its supervised subsystems were retrained on the concatenation of the train and development data.

4 Experiments

In all experiments, systems were trained while omitting debatable examples with scores of 2 as suggested by the task organizers. The development set was used both to fit the hyperparameters (ablations, lambdas) and the eventual ensemble.

String Similarity Ablations The MT evaluation metrics and string similarities contributed varying amounts to that system. In Table 1 we show the score achieved by the logistic regression system built using just that one measure (in the *Factored* column) as well as the F1 achieved by the logistic regression when only that one measure is left out (*Ablated* column). BLEU was omitted from the subsystem as a result of this analysis.

Ensemble Construction We focused our ensembles only on the output of our individual components, ignoring the features from the original data they attempt to model. Table 3 shows the weights of these components. Note that NormalizedAvg produced larger outputs than the rest; as a result its coefficient is about 10 times smaller than its effect.

System	Pearson	F1	maxF1
MITRE	**61.9**	66.7	**71.6**
RTM-DCU	57.0	54.0	69.1
HLTC-UST	56.3	65.1	67.6
ASOBEK	50.4	**67.2**	66.3
MITRE components			
RNN	**60.8**		**71.8**
Paris	58.7		68.2
RandProj	54.9		64.6
SimMat_avg	54.6		64.7
SimMat_minmax	53.5		62.8
Aligner	51.8		61.9
NormalizedAvg	45.8		61.1

Table 2: Test scores of Semantic Similarity Systems (%).

5 Results

The evaluation of our components on the competition test set is shown in Table 2, along with a sample of top-scoring competitors. Our best ensemble achieves 0.619 Pearson correlation with expert judgments, a state-of-the-art result. In contrast, the correlation of crowdsourced annotations with expert ratings is 0.735 (Xu et al., 2015). Our system's F1 on the binary paraphrase judgment task was 0.667, with a maximum F1 of 0.716 using an optimal threshold. Additionally several individual components performed well in isolation. The recurrent neural network alone achieved Pearson of 0.608 and a max F1 of 0.718.

6 Conclusion

Seven models of semantic similarity were combined for paraphrase detection in Twitter. This ensemble placed first in the Semantic Similarity competition organized within SemEval-2015 Task 1. The similarity judgments showed 0.619 correlation with expert judgment, a relative improvement of 8.6% over other published results (Xu et al., 2015).

Our best performing single system represents a novel departure from existing paraphrase detection approaches. The recurrent neural network makes use of the relationships between aligned word pairs, an approach which we feel is well-suited to informal contexts where explicit models of syntax face additional challenges.

Component	Φ	Component	Φ
RNN1	−1.89	SimMat_minmax	0.84
RNN2	−1.11	Aligner	0.28
Paris	−1.81	NormalizedAvg	−0.034
SimMat_avg	−1.28	bias	0.91
RandProj	1.11		

Table 3: Final MITRE component coefficients in the ensemble logistic regression.

Acknowledgments

This work was funded under the MITRE Innovation Program. Many thanks to John Burger for his comments on machine translation alignments. Approved for Public Release; Distribution Unlimited: Case Number 15-0811.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, I. Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David 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. Available at `http://lpsolve.sourceforge.net/5.5/`.

Moses S. Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388.

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*, COLING '04.

Jeffrey L. Elman. 1990. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211.

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*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, `http://arxiv.org/abs/1207.0580`.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proc. of the Ninth MT Summit*, pages 240–247.

Vladimir Iosifovich 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)*, August.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations Workshop*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

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, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to Twitter. In *Human Language Technologies: The*

2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 181–189.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D. Manning, and Andrew Y. Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Ronald J. Williams and David Zipser. 1990. Gradient-based learning algorithms for recurrent connectionist networks. pages 433–486.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.