Talla at SemEval-2017 Task 3: Identifying Similar Questions Through Paraphrase Detection

Byron V. Galbraith, Bhanu Pratap, Daniel Shank
Talla
Boston, MA, USA
{byron, bhanu, daniel}@talla.com

Abstract

This paper describes our approach to the SemEval-2017 shared task of determining question-question similarity in a community question-answering setting (Task 3B). We extracted both syntactic and semantic similarity features between candidate questions, performed pairwise-preference learning to optimize for ranking order, and then trained a random forest classifier to predict whether the candidate questions were paraphrases of each other. This approach achieved a MAP of 45.7% out of max achievable 67.0% on the test set.

1 Introduction

A large amount of information of interest to users of community forums is stored in semi-structured text, but surfacing that information can be challenging given the variety of ways users can phrase their search queries. Question-answering is a significant task for both natural language processing (NLP) and information retrieval (IR), as both the actual terms used in the query plus the semantic intent of the query itself need to be accounted for in surfacing relevant potential answers. The Community Question Answering (cQA) task of SemEval-2017 (Nakov et al., 2017) seeks to address this problem through several related subtasks around effectively determining and ranking the relevance of related stored questions and associated answers.

We chose to focus on subtask B: question-question similarity. This problem can be seen as one of paraphrase detection – determine if two questions have the same meaning. We reviewed existing performant paraphrase detection methods and selected several to implement and ensemble (Ji and Eisenstein, 2013; Wan et al., 2006; Wang and Ittycheriah, 2015; Filice et al., 2015) along with the related question IR system rank provided in the dataset. As paraphrase detection is a classification problem while subtask B is a ranking problem, we also incorporated pairwise-preference learning (Joachims, 2002; Fürnkranz and Hüllermeier, 2003) to aid in improving the key metric of mean average precision (MAP).

The rest of the paper is organized as follows. Section 2 provides a detailed description of our system, including the key identified features that were extracted, while Section 3 provides the results from experiments used to evaluate the system. Section 4 concludes the paper with a summary of the work and directions for future exploration.

2 System Description

Our approach consisted of four parts: data preparation, feature extraction, pairwise-preference learning, and paraphrase classification. All code was implemented in Python 3.5. For data extraction, we converted the XML documents provided by (Nakov et al., 2017) into pandas DataFrames, retaining the subject text, body text, and metadata related to the original and related questions. The feature extraction and the pairwise-preference learning phase are described below. Classification was handled with a random forest classifier containing 2000 weak estimators.

2.1 Feature Extraction

We computed features as described in several leading paraphrase detection method papers. One of which, fine-grained textual features (Wan et al., 2006), failed to produce any significant value during further evaluation for this task and so were discarded. In addition to the paraphrase detection features, we also incorporated the reciprocal of the

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 375–379,
Vancouver, Canada, August 3 - 4, 2017.

reported IR system rank of the related question as an additional feature.

Unless otherwise noted, question texts for feature extraction were created by concatenating the subject and body fields of the question, all terms were made lowercase, and stop words were removed.

2.1.1 Tree Kernels

Tree kernel (TK) features (Filice et al., 2015) were derived by generating parse trees of the two sentences, then defining a kernel that allows for a numerical distance to be computed. The kernel takes all possible valid (not necessarily terminal) partial tree structures within the sentence parse trees and counts the amount of overlap between the two. The result is a score for every pair of sentences.

The kernel function $K(S_1, S_2)$ for two trees S_1 and S_2 is defined as follows:

$$K(S_1, S_2) = \sum_{n_1 \in N_{S_1}} \sum_{n_2 \in N_{S_2}} \Delta(n_1, n_2)$$

where $\Delta(n_1, n_2)$ is the Partial Tree Kernel (PTK) function as defined in (Filice et al., 2015). A standard kernel norm is then applied, given by:

$$\frac{K(S_1, S_2)}{\sqrt{K(S_1, S_1)K(S_2, S_2)}}.$$

We computed distances for both constituency trees and dependency trees. For constituency parse trees, words that occur in both sentences were marked along with their part of speech in order to increase the effect of shared terms belonging to similar subtrees. Dependency parse trees, on the other hand, were constructed so that non-leaf nodes are made up entirely of dependency types (rather than parts of speech). For example a single ROOT node may have nodes *nsubj* and *dobj* as children. Leaves were all tokens representing words themselves, and every interior node had a child that was a leaf. The final features produced were the result of the kernel applied to the constituency parse tree and that result multiplied by the result from the kernel applied to the dependency parse tree.

2.1.2 TF-KLD

TF-KLD (Term Frequency Kullback-Leibler Divergence) (Ji and Eisenstein, 2013) is a supervised TF weighting scheme based on modeling probability distributions of phrases being aligned with or without the presence of a particular term. More formally:

We assume labeled sentence pairs $\langle \vec{w}_i^{(2)}, \vec{w}_i^{(2)}, r_i \rangle$, where $\vec{w}_i^{(1)}$ is the binarized vector of bigram and unigram occurrence for the first sentence, $\vec{w}_i^{(2)}$ is the bigram and unigram occurrence vector for the second, and $r_i \in \{0, 1\}$ is an indicator of whether the two sentences match. We assume the order of the sentences are irrelevant, and for each feature with index k we define two Bernoulli distributions:

$$p_k = P(w_{ik}^{(1)} | w_{ik}^{(2)}, r_i = 1)$$

which is the probability that feature k appears in the first sentence given that k appears in the second and both are matched, and

$$q_k = P(w_{ik}^{(1)} | w_{ik}^{(2)}, r_i = 0)$$

which is the probability that feature k appears in the first sentence given that k appears in the second and both are not matched.

The Kullback-Leibler divergence is a pre-metric over probability distributions, defined as $KL(p_k||q_k) = \sum_x p_k(x) \log \frac{p_k(x)}{q_k(x)}$. We calculate a KLD score for each feature k, then use this to weight the vector of non-binarized occurrences. The sparse TF-KLD vector then undergoes dimensionality reduction by means of rank-100 nonnegative matrix factorization. Finally, the cosine similarity of individual vectors is taken to give a single feature for each pair of sentences.

2.1.3 Semantic Word Alignment

Semantic word alignment (WA) (Wang and Ittycheriah, 2015) used word embeddings to infer semantic similarity between documents at the individual word level. For embeddings we used the pre-trained 300-dimensional GloVe vectors (Pennington et al., 2014).

Given a source question Q and reference question R, let $Q = \{q_0, q_1, ..., q_m\}$ and $R = \{r_0, r_1, ..., r_n\}$ denote the words in each question text. First, the cosine-similarity between all pairs of the words (q_i, r_j) was computed to form a similarity matrix (Figure 1). Next we denote the word alignment position for each query word q_i as $align_i$, similarity score as sim_i, and the inverse document frequency as idf_i. Word alignment position $align_i$ for a query word q_i in Q w.r.t words in R is equal to the position of a word r_j in R at

	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
	Talla-constrastive1	46.50	82.15	49.61	30.39	76.07	43.43	63.30
	Talla-contrastive2	46.31	81.81	49.14	29.88	74.23	42.61	62.95
4	**Talla-primary**	**45.70**$_4$	**81.48**$_2$	**49.55**$_5$	**29.59**$_9$	**76.07**$_8$	**42.61**$_8$	**62.05**$_8$
	Baseline 1 (IR)	41.85	77.59	46.42	-	-	-	-
	Baseline 2 (random)	29.81	62.65	33.02	18.72	75.46	30.00	34.77
	Baseline 3 (all 'true')	-	-	-	18.52	100.00	31.26	18.52
	Baseline 3 (all 'false')	-	-	-	-	-	-	81.48

Table 1: System performance on the SemEval-2017 test dataset

which q_i has maximum similarity score sim_i. Finally, we compute a set of distinct word alignment features as:

- **similarity:** $f_0 = \sum_i sim_i * idf_i / \sum_i idf_i$. This feature represents question similarity based on the aligned words.
- **dispersion:** $f_1 = \sum_i (|align_i - align_{i-1} - 1|)$. This feature is a measure of contiguously aligned words.
- **penalty:** If we denote the position of unaligned words (where $sim_i = 0$) as $unalign_i$, then this feature penalizes pairs with unaligned question words and was calculated as $f_2 = \sum_{unalign_i} idf_i / \sum_i idf_i$.
- **five important words:** $f_{i_{th}} = sim_{i_{th}} * idf_{i_{th}}$. This feature set included the similarity score of the top five important words in the question text, where importance of a word was based on its IDF score.

The first three features were computed in both directions i.e. for (Q_i, R_j) and (R_j, Q_i). The cosine similarity of the aggregate of all embeddings in the questions was also computed. This process was repeated separately for both question subjects and bodies (instead of on the combined concatenated text) for a total of 24 distinct features.

2.2 Pairwise-Preference Learning

Since the official evaluation metric for Subtask B was MAP, we adopted a ranking approach to indirectly optimize for MAP. Given an original question Q_i and its list of corresponding related questions $\{R_1, R_2, ..R_{10}\}$, we are interested in learning a ranking of this list, where relevant questions are ranked higher than irrelevant ones. An alternative way to learn this ranking is to classify if a pair from a set of pairs formed within one group,

	r_1	r_2	r_3	r_4	r_5	r_6
q_1	0.2	0.7	0	0	0	0.4
q_2	0	0.1	0.4	0.2	0	0
q_3	0.3	0.2	0	0	0.5	0

	r_1	r_2	r_3	r_4	r_5	r_6
q_1		X				
q_2			X			
q_3					X	

Figure 1: Word alignment matrix example. The upper table contains the cosine similarity scores between words in questions Q and R, while the lower table contains the corresponding word-alignment.

where a group is formed for each original question Q_i is correctly ordered or not. This principle is called "pairwise-preference learning" (Joachims, 2002; Fürnkranz and Hüllermeier, 2003).

To make use of this approach we transformed the datasets from question-question(or question-comment) pairs into a set of instance pairs. That is, we presented a pair of answers with one correct and one incorrect answer to the same question. Number of features were kept constant, while feature values were equal to the difference between the values of two answers in the instance pair.

In training phase, for each question group $(Q_i, \{R_1, R_2, ..R_{10}\})$ we generated labeled pairs as "correct-pair(Q_i, R_j) minus incorrect-pair(Q_i, R_k)" with label *true* and "incorrect-pair(Q_i, R_k) minus correct-pair(Q_i, R_j)" with label *false*. In this way, we generated $2 * (n_c + n_i)$ instance pairs for each question group, where n_c and n_i is the number of correct pairs and number of incorrect pairs within a group respectively.

In testing phase, number of instance pairs generated for a question group$(Q_i, \{R_1, R_2, ..R_{10}\})$ were equal to the number of all possible pairs

within that question group. Then, our model assigned a probability to each of these instance pairs that it is correctly ordered. To create a final score for each related-question R_j, we took the sum of probabilities over all pairs in which R_j was ranked first. This final score was then used to create a ranked list of related-questions R_j for each original question Q_i.

3 Experiments and Evaluation

We combined the provided training and dev datasets as our system training set and used the provided SemEval-2016 test data with gold labels as our test set. No additional external data, other than pre-trained word embeddings, were used. We evaluated different classifier hyperparameters using 10-fold cross-validation and ultimately chose a random forest classifier with 2000 trees as our final model.

This system achieved fourth place overall (Table 1) on the SemEval-2017 test dataset, and while both contrastive submissions placed higher than the primary, nether was able to achieve a greater MAP than the third place entry. Contrastive1 was identical in feature set to the primary submission, but included the SemEval-2016 test dataset as part of the training data, suggesting that MAP can be improved by increasing the amount of examples used to train the system. Contrastive2 did not include the extra data and also omitted the TF-KLD features. Comparing the effects of ablating the other individual features (Table 2) across both SemEval-2016 and SemEval-2017 test datasets demonstrated that both the TF-KLD and TK features were minimally effective. The IR system features had a dramatic difference between the two years – in 2016 it accounted for a 0.022 gain in MAP, while in 2017 it produced a 0.010 reduction. In both cases the WA features contributed the most, with gains of 0.041 and 0.034, respectively.

Subtask B of Task 3 combines the PerfectMatch and Relevant classes into a single positive class for purposes of evaluation. Given that this approach treated question-question similarity as a paraphrase detection problem, the expectation was that this model would do better on the PerfectMatch and Irrelevant samples, but have a harder time with Relevant questions. This is seen in the SemEval-2016 data (Figure 2), where there is good separation between the computed pairwise-preference scores of Irrelevant and PerfectMatch

	MAP	
Features	**2016**	**2017**
Max	0.886	0.670
All features	0.781	0.457
All - TK	0.775	0.452
All - TF-KLD	0.773	0.464
All - IR	0.759	0.467
All - WA	**0.740**	**0.423**
Baseline 1 (IR)	0.748	0.419
Baseline 2 (random)	0.470	0.298

Table 2: Ablation studies of the four ensembled feature sources against SemEval-2016 and SemEval-2017 test data. Bolded values indicate the largest loss due to ablation.

samples while the Relevant class is spread evenly between the other two. Surprisingly, this dynamic changed when applied to the SemEval-2017 data, resulting in improved separation for the Relevant class, but worse for both Irrelevant and PerfectMatch classes.

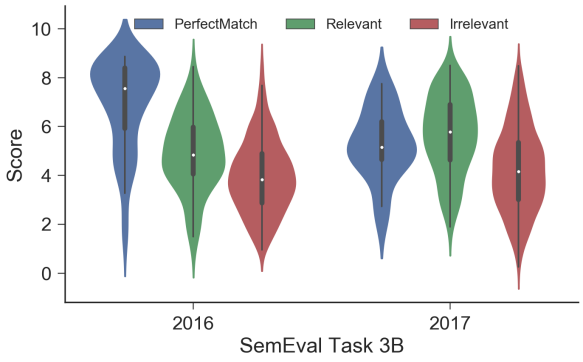

Figure 2: Our model was unable to consistently score PerfectMatch class questions over Irrelevant ones across SemEval datasets, suggesting that it overfit to the distribution of the training data.

Both the significant swing in IR feature contribution and drop in ability to detect PerfectMatch samples as positive examples of question-question similarity are reflected in the change in makeup of the dataset (Table 3). The train + dev dataset we used for general training was more closely aligned with the distribution of class labels in 2016 than in 2017, suggesting a potential i.i.d. data dependence on this approach to produce good results on test data.

Dataset	n	PM	R	I
train	2669	0.09	0.32	0.59
dev	500	0.12	0.31	0.57
test-2016	700	0.11	0.22	0.67
test-2017	880	0.03	0.16	0.81

Table 3: Distribution of the PerfectMatch (PM), Relevant (R), and Irrelevant (I) classes within the datasets.

4 Summary

We described a system that relies on an ensemble of syntactic, semantic, and IR features to detect question-question similarity and demonstrated it on the SemEval-2017 community question answering shared task. Of the four feature sources we evaluated, the semantic word alignment features provided the largest contributed and consistent boost in MAP. Features derived from TF-KLD and tree kernel methods had modest effects. The efficacy of the IR-derived features varied from providing a noticeable gain on historical data vs a significant drop on the current test set, likely attributable to the significant increase in the number of Irrelevant class samples. Future work will explore how to compensate for highly unbalanced class scenarios.

References

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2015. Structural representations for learning relations between pairs of texts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 1003–1013. http://aclweb.org/anthology/P/P15/P15-1097.pdf.

Johannes Fürnkranz and Eyke Hüllermeier. 2003. Pairwise preference learning and ranking. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and L. Todorovski, editors, *Proceedings of the 14th European Conference on Machine Learning (ECML-03)*. Springer-Verlag, Cavtat, Croatia, volume 2837 of *Lecture Notes in Artificial Intelligence*, pages 145–156. http://www.ke.tu-darmstadt.de/ juffi/publications/ecml-03.pdf.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 891–896. http://aclweb.org/anthology/D/D13/D13-1090.pdf.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '02, pages 133–142. https://doi.org/10.1145/775047.775067.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Vancouver, Canada, SemEval '17.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the "para-farce" out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop*. volume 2006.

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *CoRR* abs/1507.02628. http://arxiv.org/abs/1507.02628.