SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter (PIT)

Wei Xu and Chris Callison-Burch
University of Pennsylvania
Philadelphia, PA, USA
xwe, ccb@cis.upenn.edu

William B. Dolan
Microsoft Research
Redmond, WA, USA
billdol@microsoft.com

Abstract

In this shared task, we present evaluations on two related tasks Paraphrase Identification (PI) and Semantic Textual Similarity (SS) systems for the Twitter data. Given a pair of sentences, participants will be asked to produce a binary yes/no judgement or a graded score to measure their semantic equivalence. The task features a newly constructed Twitter Paraphrase Corpus that contains 18,762 sentence pairs. A total of 19 teams participated, submitting 36 runs to the PI task and 26 runs to the SS task. The evaluation shows encouraging results and open challenges for future research. The best systems scored a F1-measure of 0.674 for the PI task and a Pearson correlation of 0.619 for the SS task respectively, comparing to a strong baseline using logistic regression model of 0.589 F1 and 0.511 Pearson; and the best SS systems can often reach >0.80 Pearson on well-formed text. This shared task also provides insights into the relation between the PI and SS tasks and suggests the importance to bringing these two research areas together. We make all the data, baseline systems and evaluation scripts publicly available.[1]

1 Introduction

The ability to identify paraphrases, i.e. alternative expressions of the same (or similar) meaning, and the degree of their semantic similarity has proven useful for a wide variety of natural language processing applications (Madnani and Dorr, 2010). It is particularly useful to overcome the challenge of high redundancy in Twitter and the sparsity inherent in their short texts (e.g. *oscar nom'd doc* ↔ *Oscar-nominated documentary*; *some1 shot a cop* ↔ *someone shot a police*;). Emerging research shows paraphrasing techniques applied to Twitter data can improve tasks like on first story detection (Petrović et al., 2012), information retrieval (Zanzotto et al., 2011) and text normalization (Xu et al., 2013; Wang et al., 2013).

Previously, many researchers have investigated ways of automatically detecting paraphrases on more formal texts, like newswire text. The ACL Wiki[2] gives an excellent summary of the state-of-the-art paraphrase identification techniques. These can be categorized into supervised methods (Qiu et al., 2006; Wan et al., 2006; Das and Smith, 2009; Socher et al., 2011; Blacoe and Lapata, 2012; Madnani et al., 2012; Ji and Eisenstein, 2013) and unsupervised methods (Mihalcea et al., 2006; Rus et al., 2008; Fernando and Stevenson, 2008; Islam and Inkpen, 2007; Hassan and Mihalcea, 2011). A few recent studies have highlighted the potential and importance of developing paraphrase identification (Zanzotto et al., 2011; Xu et al., 2013) and semantic similarity techniques (Guo and Diab, 2012) specifically for tweets. They also indicated that the very informal language, especially the high degree of lexical variation, used in social media has posed serious challenges to both tasks.

[1] http://www.cis.upenn.edu/~xwe/semeval2015pit/

[2] http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art)

Paraphrase?	Sentence 1	Sentence 2
yes	Ezekiel Ansah wearing 3D glasses wout the lens	Wait Ezekiel ansah is wearing 3d movie glasses with the lenses knocked out
yes	Marriage equality law passed in Rhode Island	Congrats to Rhode Island becoming the 10th state to enact marriage equality
yes	Aaaaaaaaand stephen curry is on fire	What a incredible performance from Stephen Curry
no	Finally saw the Ciara body party video	ciara s Body Party video is on point
no	Now lazy to watch Manchester united vs arsenal	Early lead for Arsenal against Manchester United

Table 1: Representative examples from PIT-2015 Twitter Paraphrase Corpus

	# Unique Sent	# Sent Pair	# Paraphrase	# Non-Paraphrase	# Debatable
Train	13231	13063	3996 (30.6%)	7534 (57.7%)	1533 (11.7%)
Dev	4772	4727	1470 (31.1%)	2672 (56.5%)	585 (12.4%)
Test	1295	972	175 (18.0%)	663 (68.2%)	134 (13.8%)

Table 2: Statistics of PIT-2015 Twitter Paraphrase Corpus. The percentage of paraphrases is lower in the test set because it was constructed without topic selection.

The SemEval-2015 shared task on **P**araphrase and Semantic Similarity **I**n **T**witter (PIT) uses a training and development set of 17,790 sentence pairs and a test set of 972 sentence pairs with paraphrase and semantic similarity annotations (see examples in Table 1) that we developed earlier in (Xu, 2014) and (Xu et al., 2014). This PIT-2015 paraphrase dataset is distinct from the data used in previous studies in many aspects: (i) it contains sentences that are opinionated and colloquial, representing realistic informal language usage; (ii) it contains paraphrases that are lexically diverse; and (iii) it contains sentences that are lexically similar but semantically dissimilar. It raises many interesting research questions and could lead to a better understanding of our daily used language and how semantics can be captured in such language. We believe that such a common testbed will facilitate docking of the different approaches for purposes of comparison, lead to a better understanding of how semantics are conveyed in natural language, and help advance other NLP techniques for noisy user-generated text in the long run.

2 Task Description and Evaluation Metrics

The task has two sentence-level sub-tasks: a paraphrase identification task and an optional semantic textual similarity task. The two sub-tasks share the same data but differ in annotation and evaluation.

Task A – Paraphrase Identification

Given two sentences, determine whether they express the same or very similar meaning. Following the literature on paraphrase identification, we evaluate system performance by the F-1 score (harmonic mean of precision and recall) against human judgements.

Task B – Semantic Textual Similarity

Given two sentences, determine a numerical score between 0 (no relation) and 1 (semantic equivalence) to indicate their semantic similarity. Following the literature, the system outputs are compared by Pearson correlation with human scores. We also compute the maximum F-1 score as an additional data point.

3 Corpus

In this shared task, we use the Twitter Paraphrase Corpus that we first presented in (Xu, 2014) and (Xu et al., 2014). Table 2 shows the basic statistics of the corpus. The sentences are pre-processed tokenization,[3] POS and named entity tags.[4] The training and development set consists of 17,790 sentence pairs posted between April 24th and May 3rd, 2013 from 500+ trending topics featured on Twitter (excluding hashtags). The training and development set is a random split. Each sentence pair is annotated by 5 different crowdsourcing workers. For the test set, we obtain both crowdsourced and expert labels on 972 sentence pairs from 20 randomly sampled Twitter trending topics between May 13th and June 10th, 2013. We use expert labels in this SemEval evaluation. Our dataset is more realistic and balanced, containing about 70% non-paraphrases vs. the 34% non-paraphrases in the benchmark Microsoft Paraphrase Corpus derived from news articles by Dolan et al. (2004). As noted in (Das and Smith, 2009), the lack of natural non-paraphrases in the MSR corpus creates bias towards certain models.

4 Annotation

In this section, we describe our data collection and annotation methodology. Since Twitter users are free to talk about anything regarding any topic, a random pair of sentences about the same topic has a low chance of expressing the same meaning (empirically, this is less than 8%). This causes two problems: a) it is expensive to obtain paraphrases via manual annotation; b) non-expert annotators tend to loosen the criteria and are more likely to make false positive errors. To address these challenges, we design a simple annotation task and introduce two selection mechanisms to select sentences which are more likely to be paraphrases, while preserving diversity and representativeness.

4.1 Raw Data from Twitter

We crawl Twitter's trending topics and their associated tweets using public APIs.[5] According to Twitter, trends are determined by an algorithm which identifies topics that are immediately popular, rather than those that have been popular for longer periods of time or which trend on a daily basis. We tokenize, remove emoticons[6] and split tweet into sentences.

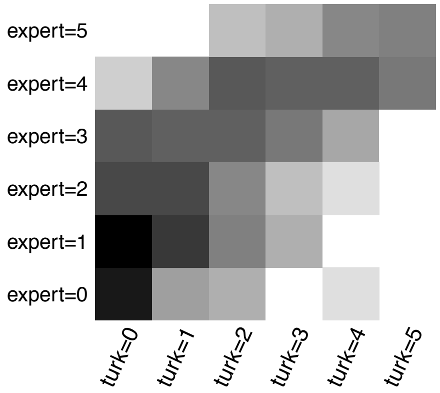

Figure 1: A heat-map showing overlap between expert and crowdsourcing annotation. The intensity along the diagonal indicates good reliability of crowdsourcing workers for this particular task; and the shift above the diagonal reflects the difference between the two annotation schemas. For crowdsourcing (turk), the numbers indicate how many annotators out of 5 picked the sentence pair as paraphrases; 0,1 are considered non-paraphrases; 3,4,5 are paraphrases. For expert annotation, all 0,1,2 are non-paraphrases; 4,5 are paraphrases. Medium-scored cases (2 for crowdsourcing; 3 for expert annotation) are discarded in the system evaluation of the PI sub-task.

4.2 Task Design on Mechanical Turk

We show the annotator an **original** sentence, then ask them to pick sentences with the same meaning from 10 **candidate** sentences. The original and candidate sentences are randomly sampled from the same topic. For each such 1 vs. 10 question, we obtain binary judgements from 5 different annotators, paying each annotator $0.02 per question. On average, each question takes one annotator about 30 $\sim$ 45 seconds to answer.

4.3 Annotation Quality

We remove problematic annotators by checking their Cohen's Kappa agreement (Artstein and Poesio, 2008) with other annotators. We also compute inter-annotator agreement with an expert annotator on the test dataset of 972 sentence pairs. In the expert annotation, we adopt a 5-point Likert scale to measure the degree of semantic similarity between

[3] The tokenizer was developed by O'Connor et al. (2010): https://github.com/brendano/tweetmotif

[4] The POS tagger was developed by Derczynski et al. (2013) and the NER tagger was developed by Ritter et al. (2011): https://github.com/aritter/twitter_nlp

[5] More information about Twitter's APIs: https://dev.twitter.com/docs/api/1.1/overview

[6] We use the toolkit developed by O'Connor et al. (2010): https://github.com/brendano/tweetmotif

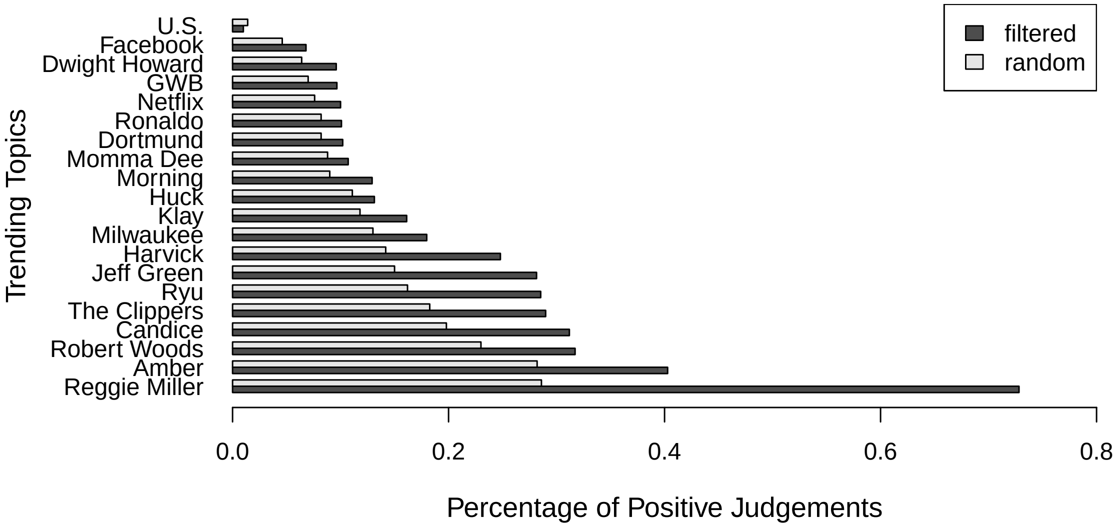

Figure 2: The proportion of paraphrases (percentage of positive votes from annotators) vary greatly across different topics. Automatic filtering in Section 4.4 roughly doubles the paraphrase yield.

sentences, which is defined by Agirre et al. (2012) as follows:

5: Completely equivalent, as they mean the same thing;
4: Mostly equivalent, but some unimportant details differ;
3: Roughly equivalent, but some important information differs/missing.
2: Not equivalent, but share some details;
1: Not equivalent, but are on the same topic;
0: On different topics.

Although the two scales of expert and crowdsourcing annotation are defined differently, their Pearson correlation coefficient reaches 0.735 (two-tailed significance 0.001). Figure 1 shows a heatmap representing the detailed overlap between the two annotations. It suggests that the graded similarity annotation task could be reduced to a binary choice in a crowdsourcing setup. As for the binary paraphrase judgements, the integrated judgement of five crowdsourcing workers achieve a F1-score of 0.823, precision of 0.752 and recall of 0.908 against expert annotations.

4.4 Automatic Summarization Inspired Sentence Filtering

We filter the sentences within each topic to select more probable paraphrases for annotation. Our method is inspired by a typical problem in extractive summarization, that the salient sentences are likely redundant (paraphrases) and need to be removed in the output summaries. We employ the scoring method used in SumBasic (Nenkova and Vanderwende, 2005; Vanderwende et al., 2007), a simple but powerful summarization system, to find salient sentences. For each topic, we compute the probability of each word $P(w_i)$ by simply dividing its frequency by the total number of all words in all sentences. Each sentence s is scored as the average of the probabilities of the words in it, i.e.

$$Salience(s) = \sum_{w_i \in s} \frac{P(w_i)}{|\{w_i | w_i \in s\}|} \qquad (1)$$

We then rank the sentences and pick the **original** sentence randomly from top 10% salient sentences and **candidate** sentences from top 50% to present to the annotators.

In a trial experiment of 20 topics, the filtering technique double the yield of paraphrases from 152 to 329 out of 2000 sentence pairs over naïve random sampling (Figure 2 and Figure 3). We also use PINC (Chen and Dolan, 2011) to measure the quality of paraphrases collected (Figure 4). PINC was designed to measure n-gram dissimilarity between two sentences, and in essence it is the inverse of BLEU. In general, the cases with high PINC scores include more complex and interesting rephrasings.

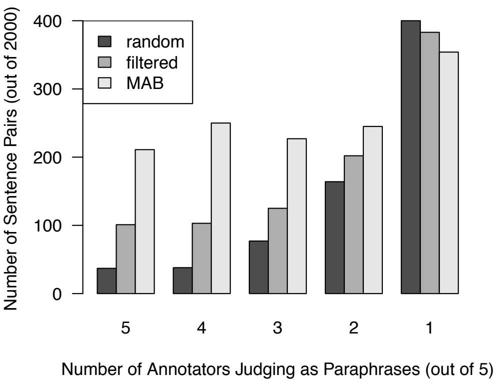

Figure 3: Numbers of paraphrases collected by different methods. The annotation efficiency (3,4,5 are regarded as paraphrases) is significantly improved by the sentence filtering and Multi-Armed Bandits (MAB) based topic selection.

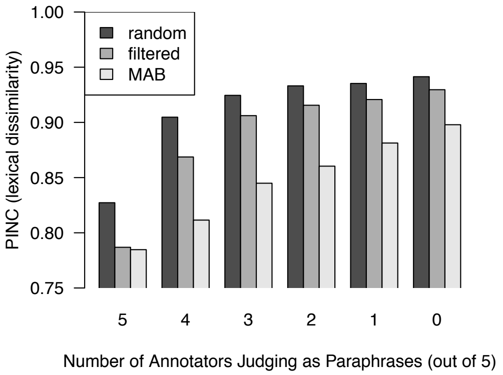

Figure 4: PINC scores of paraphrases collected. The higher the PINC, the more significant the rewording. Our proposed annotation strategy quadruples paraphrase yield, while not greatly reducing diversity as measured by PINC.

4.5 Topic Selection using Multi-Armed Bandits (MAB) Algorithm

Another approach to increasing paraphrase yield is to choose more appropriate topics. This is particularly important because the number of paraphrases varies greatly from topic to topic and thus the chance to encounter paraphrases during annotation (Figure 2). We treat this topic selection problem as a variation of the Multi-Armed Bandit (MAB) problem (Robbins, 1985) and adapt a greedy algorithm, the bounded ϵ-first algorithm, of Tran-Thanh et al. (2012) to accelerate our corpus construction.

Our strategy consists of two phases. In the first exploration phase, we dedicate a fraction of the total budget, ϵ, to explore randomly chosen arms of each slot machine (trending topic on Twitter), each m times. In the second exploitation phase, we sort all topics according to their estimated proportion of paraphrases, and sequentially annotate $\lceil \frac{(1-\epsilon)B}{l-m} \rceil$ arms that have the highest estimated reward until reaching the maximum $l = 10$ annotations for any topic to insure data diversity.

We tune the parameters m to be 1 and ϵ to be between 0.35 $\sim$ 0.55 through simulation experiments, by artificially duplicating a small amount of real annotation data. We then apply this MAB algorithm in the real-world. We explore 500 random topics and then exploited 100 of them. The yield of paraphrases rises to 688 out of 2000 sentence pairs by using MAB and sentence filtering, a 4-fold increase compared to only using random selection (Figure 3).

5 Baselines

We provide three baselines, including a random baseline, a strong supervised baseline and a state-of-the-art unsupervised system:

Random:
This baseline provides a randomized real number between [0, 1] for each test sentence pair as semantic similarity score, and uses 0.5 as cutoff for binary paraphrase identification output.

Logistic Regression:
This is a supervised logistic regression (LR) baseline used by Das and Smith (2009). It uses simple n-gram (also in stemmed form) overlapping features but shows very competitive performance on the MSR news paraphrase corpus. It uses 0.5 as cutoff to create binary outputs for the paraphrase identification task.

Weighted Matrix Factorization (WTMF):[7]
The second baseline is a state-of-the-art unsupervised method developed by Guo and Diab (2012). It is specially developed for short sentences by modeling the semantic space of both words that are present in and absent from the sentences (Guo and Diab, 2012). The model

[7] The source code and data for WTMF is available at: http://www.cs.columbia.edu/~weiwei/code.html

was learned from WordNet (Fellbaum, 2010), OntoNotes (Hovy et al., 2006), Wiktionary, the Brown corpus (Francis and Kucera, 1979). It uses 0.5 as cutoff in the binary paraphrase identification task.

6 Systems and Results

A total of 18 teams participated in the PI task (required), 13 of which also submitted to the SS task (optional). Every team submitted 2 runs except one (up to 2 were are allowed).

6.1 Evaluation Results

Table 3 shows the evaluation results. We use the F1-score and Pearson correlation as the primary evaluation metric for the PI and SS task respectively. The results are very exciting that most systems outperformed the two strong baselines we chose, while still showing room for improvement towards the human upper-bound estimated by the crowdsourcing worker's performance.

6.2 Discussion

Most participants choose supervised methods, except for MathLingBp who uses semi-supervised, Columbia and Yamraj who use unsupervised methods. While the best performed systems are supervised, the best unsupervised system still outperforms some supervised systems and the state-of-the-art unsupervised baseline. About half of systems use word embeddings and many use neural networks.

To out best knowledge, this is the first time to have a large number of systems in an evaluation that has the two related tasks — paraphrase identification and semantic similarity, side by side for comparison. One interesting observation that comes out is the performance of the same system on the two tasks ("F1 vs. Pearson") are not necessarily related. For example, ASOBEK ranked 1st (out of 35 runs) and 18th (out of 25 runs) in the PI and SS tasks respectively, RTM-DCU ranked 27th and 3rd, while the MITRE system ranked 3nd and 1st place. Neither "F1 vs. max-F1" nor "Pearson vs. maxF1" nor "F1 vs. Pearson" show a strong correlation. It implies that (i) high-performance PI systems can be developed focusing on the binary classification problem without focusing on the degree of similarity; (ii) it is crucial to select the threshold to balance precision and recall for the PI binary classification problem; (iii) it is important for SS system to handle the debatable cases proporiately.

6.3 Participants' Systems

There are in total 19 teams participated:

AJ: This team utilizes TERp and BLEU – automatic evaluation metrics for Machine Translation. The system uses a logistic regression model and performs threshold selection.

AMRITACEN: This team uses Recursive Auto Encoders (RAEs). The matrix generated for the given input sentences is of variable size, then converted to equal sized matrix using repeat matrix concept.

ASOBEK: This team uses SVM classifier with simple lexical word overlap and character n-grams features.

CDTDS: This team uses support vector regression trained only on the training set using the numbers of positive votes out of the 5 crowdsourcing annotations. Outputs are thresholded at 0.35 for binary predictions.

Columbia: This system maps each original sentence to a low dimensional vector as Orthogonal Matrix Factorization (Guo et al., 2014), and then computes similarity score based on the low dimensional vectors.

Depth: This team uses neural network that learns representation of sentences, then compute similarity scores based on hidden vector representations between two sentences.

EBIQUITY: This team trains supervised SVM and logistic regression models using features of semantic similarities between sentence pairs.

ECNU: This team adopts typical machine learning classifiers and uses a variety of features, such as surface text, semantic level, textual entailment, word distributional representations by deep learning methods.

FBK-HLT: This team uses supervised learning model with different features for the 2 runs,

Rank				**Paraphrase Identification (PI)**			**Semantic Similarity (SS)**			
PI	**SS**	**Team**	**Run**	**F1**	**Precision**	**Recall**	**Pearson**	**maxF1**	**mPrec**	**mRecall**
		Human Upperbound		**0.823**	**0.752**	**0.908**	**0.735**	——	——	——
1		ASOBEK	01_svckernel	0.674^{1}	0.680	0.669	0.475^{18}	0.616	0.732	0.531
	8	ASOBEK	02_linearsvm	0.672^{2}	0.682	0.663	0.504^{14}	0.663	0.723	0.611
2	1	MITRE	01_ikr	0.667^{3}	0.569	0.806	0.619^{1}	0.716	0.750	0.686
3		ECNU	02_nnfeats	0.662^{4}	0.767	0.583	——	——	——	——
4		FBK-HLT	01_voted	0.659^{5}	0.685	0.634	0.462^{19}	0.607	0.551	0.674
5		TKLBLIIR	02_gs0105	0.659^{5}	0.645	0.674	——	——	——	——
		MITRE	02_bieber	0.652^{7}	0.559	0.783	0.612^{2}	0.724	0.753	0.697
6		HLTC-HKUST	02_run2	0.652^{7}	0.574	0.754	0.545^{6}	0.669	0.738	0.611
	3	HLTC-HKUST	01_run1	0.651^{9}	0.594	0.720	0.563^{5}	0.676	0.697	0.657
		ECNU	01_mlfeats	0.643^{10}	0.754	0.560	——	——	——	——
7	4	AJ	01_first	0.622^{11}	0.523	0.766	0.527^{7}	0.642	0.571	0.731
8	5	DEPTH	02_modelx23	0.619^{12}	0.652	0.589	0.518^{8}	0.636	0.602	0.674
9	9	CDTDS	01_simple	0.613^{13}	0.547	0.697	0.494^{15}	0.626	0.675	0.583
		CDTDS	02_simplews	0.612^{14}	0.542	0.703	0.491^{16}	0.624	0.589	0.663
		DEPTH	01_modelh22	0.610^{15}	0.647	0.577	0.505^{13}	0.638	0.642	0.634
	10	FBK-HLT	02_multilayer	0.606^{16}	0.676	0.549	0.480^{17}	0.604	0.504	0.754
10		ROB	01_all	0.601^{17}	0.519	0.714	0.513^{10}	0.612	0.721	0.531
11		EBIQUITY	01_run	0.599^{18}	0.651	0.554	——	——	——	——
		TKLBLIIR	01_gsc054	0.590^{19}	0.461	0.817	——	——	——	——
		EBIQUITY	02_run	0.590^{19}	0.646	0.543	——	——	——	——
		BASELINE	**logistic reg.**	$\mathbf{0.589}^{21}$	**0.679**	**0.520**	$\mathbf{0.511}^{11}$	**0.601**	**0.674**	**0.543**
12	11	COLUMBIA	02_ormf ◇	0.588^{22}	0.593	0.583	0.425^{20}	0.599	0.623	0.577
13	12	HASSY	01_train	0.571^{23}	0.449	0.783	0.405^{22}	0.645	0.657	0.634
14		RTM-DCU	01_PLSSVR	0.562^{24}	0.859	0.417	0.564^{4}	0.678	0.649	0.709
		COLUMBIA	01_ormf ◇	0.561^{25}	0.831	0.423	0.425^{20}	0.599	0.623	0.577
		HASSY	02_traindev	0.551^{25}	0.423	0.789	0.405^{22}	0.629	0.648	0.611
	2	RTM-DCU	02_SVR	0.540^{27}	0.883	0.389	0.570^{3}	0.693	0.695	0.691
		BASELINE	**WTMF ◇**	$\mathbf{0.536}^{28}$	**0.450**	**0.663**	$\mathbf{0.350}^{26}$	**0.587**	**0.570**	**0.606**
	6	ROB	02_all	0.532^{29}	0.388	0.846	0.515^{9}	0.616	0.685	0.560
	7	MATHLING	02_twimash ◇	0.515^{30}	0.364	0.880	0.511^{11}	0.650	0.648	0.651
15		MATHLING	01_twiemb ◇	0.515^{30}	0.454	0.594	0.229^{27}	0.562	0.638	0.503
16		YAMRAJ	01_google ◇	0.496^{32}	0.725	0.377	0.360^{25}	0.542	0.502	0.589
17		STANFORD	01_vs	0.480^{33}	0.800	0.343	——	——	——	——
		AJ	02_second	0.477^{34}	0.618	0.389	——	——	——	——
	13	YAMRAJ	02_lexical ◇	0.470^{35}	0.677	0.360	0.363^{24}	0.511	0.508	0.514
late	late	AMRITACEN	01_RAE	0.457	0.543	0.394	0.303	0.457	0.543	0.394
18		WHUHJP	02_whuhjp	0.425^{36}	0.299	0.731	——	——	——	——
		WHUHJP	01_whuhjp	0.387^{37}	0.275	0.651	——	——	——	——
		BASELINE	**random ◇**	$\mathbf{0.266}^{38}$	**0.192**	**0.434**	$\mathbf{0.017}^{28}$	**0.350**	**0.215**	**0.949**

Table 3: Evaluation results. The superscripts are the ranks of systems, ordered by F1 for Paraphrase Identification (PI) task and Pearson for Semantic Similarity (SS). ◇ indicates unsupervised or semi-supervised system. In total, 19 teams participated in the PI task, of which 14 teams also participated in the SS task. The first column presents the rank of each team in the two tasks. Note that although the two sub-tasks share the same test set of 972 sentence pairs, the PI task ignores 134 debatable cases (received a medium-score from expert annotator) and uses only 838 pairs (663 paraphrases and 175 non-paraphrases) in evaluation, while SS task uses all 972 pairs. This causes that the F1-score in the PI task can be higher than the maximum F1-score in the SS task. Also note that the F1-scores of the baselines in the PI task are higher than reported in the Table 2 of (Xu et al., 2014), because the later reported maximum F1-scores on the PI task, ignoring the debatable cases.

such as n-gram overlap, word alignment and edit distance.

Hassy: This team uses a bag-of-embeddings approach via supervised learning. Two sentences are first embedded into a vector space, and then the system computes the dot-product of the two sentence embeddings. This system does not rely on any unlabeled corpora and tools.

HLTC-HKUST: This team uses supervised classification with a standard two-layer neural network classifier, taking into consideration all the six labels in the objective function. The features used include translation metrics, lexical, syntactic and semantic similarity scores, the latter with an emphasis on aligned semantic roles comparison.

MathLingBp: This team implements the align-and-penalize architecture described by Han et al. (2013) with slight modifications and makes use of several word similarity metrics. One metric relies on a mapping of words to vectors built from the Rovereto Twitter N-Gram corpus, another on a synonym list built from Wiktionary's translations, while a third approach derives word similarity from concept graphs built using the 4lang lexicon and the Longman Dictionary of Contemporary English (Kornai et al., 2015).

MITRE: A recurrent neural network models semantic similarity between sentences using the sequence of symmetric word alignments that maximize cosine similarity between word embeddings. We include features from local similarity of characters, random projection, matching word sequences, pooling of word embeddings, and alignment quality metrics. The resulting ensemble uses both semantic and string matching at many levels of granularity.

RTM-DCU: This team uses referential translation machines (RTM) and machine translation performance prediction system (MTPP) for predicting semantic similarity where indicators of translatability are used as features (Biçici and Way, 2014) and instance selection for RTM is performed with FDA5 (Biçici and Yuret, 2014). RTM works as follows: $\boxed{\text{FDA5}} \rightarrow \boxed{\text{MTPP}} \rightarrow \boxed{\text{ML training}} \rightarrow \boxed{\text{predict}}$.

Rob: This system is inspired by a state-of-the-art semantic relatedness prediction system (Bjerva et al., 2014). It combines features from different parses with lexical and compositional distributional feature using a logistic regression model.

STANFORD: This team uses a supervised system using sentiment, phrase similarity matrix, and alignment features. Similarity metrics are based on vector space representation of phrases which was trained on a large corpus.

TkLbLiiR: This team uses a supervised model with about 15 comparison-based numeric features. The most important features are the distributional features weighted by the trend-specific information.

WHUHJP: This team uses the word2vec tool to train a vector model on the training data, then computes distributed representations of sentences in the test set and their cosine similarity.

Yamraj: This team uses pre-trained word and phrase vectors on Google News data set (about 100 billion words) and Wikipeida articles. The system relies on the cosine distance between vectors representing the sentences computed using open-source toolkit Gensim.

7 Conclusions and Future Work

We have presented the task definition, data annotation and evaluation results to the first Paraphrase and Semantic Similarity In Twitter (PIT) shared task. Our analysis provides some initial insights into the relation and the difference between paraphrase identification and semantic similarity problems. We make all the data, baseline systems and evaluation scripts publicly available.[8]

In the future, we plan to extend the task to allow leverage on more information from social networks, for example, providing the full tweets (and their ids) that are associated with each sentence and with each topic.

[8] https://github.com/cocoxu/SemEval-PIT2015

References

Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of SemEval*.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4).

Biçici, E. and Way, A. (2014). RTM-DCU: Referential translation machines for semantic similarity. In *Proceedings of SemEval*.

Biçici, E. and Yuret, D. (2014). Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*.

Bjerva, J., Bos, J., van der Goot, R., and Nissim, M. (2014). The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of SemEval*.

Blacoe, W. and Lapata, M. (2012). A comparison of vector-based representations for semantic composition. In *Proceedings of EMNLP-CoNLL*.

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of ACL*.

Das, D. and Smith, N. A. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP*.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of RANLP*.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING*.

Fellbaum, C. (2010). WordNet. In *Theory and Applications of Ontology: Computer Applications*. Springer.

Fernando, S. and Stevenson, M. (2008). A semantic similarity approach to paraphrase detection. *Computational Linguistics UK (CLUK) 11th Annual Research Colloquium*.

Francis, W. N. and Kucera, H. (1979). Brown corpus manual. Technical report, Brown University. Department of Linguistics.

Guo, W. and Diab, M. (2012). Modeling sentences in the latent space. In *Proceedings of ACL*.

Guo, W., Liu, W., and Diab, M. (2014). Fast tweet retrieval with compact binary codes. In *Proceedings of COLING*.

Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). UMBC EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of *SEM*.

Hassan, S. and Mihalcea, R. (2011). Semantic relatedness using salient semantic analysis. In *Proceedings of AAAI*.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: the 90% solution. In *Proceedings of HLT-NAACL*.

Islam, A. and Inkpen, D. (2007). Semantic similarity of short texts. In *Proceedings of RANLP*.

Ji, Y. and Eisenstein, J. (2013). Discriminative improvements to distributional sentence similarity. In *Proceedings of EMNLP*.

Kornai, A., Makrai, M., Nemeskey, D., and Recski, G. (2015). Extending 4lang using monolingual dictionaries. Unpublished manuscript.

Madnani, N. and Dorr, B. J. (2010). Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*.

Madnani, N., Tetreault, J., and Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. In *Proceedings of NAACL-HLT*.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of AAAI*.

Nenkova, A. and Vanderwende, L. (2005). The impact of frequency on summarization. Technical report, Microsoft Research. MSR-TR-2005-101.

O'Connor, B., Krieger, M., and Ahn, D. (2010). Tweetmotif: Exploratory search and topic summarization for Twitter. In *Proceedings of ICWSM*.

Petrović, S., Osborne, M., and Lavrenko, V. (2012). Using paraphrases for improving first story detection in news and twitter. In *Proceedings of NAACL-HLT*.

Qiu, L., Kan, M.-Y., and Chua, T.-S. (2006). Paraphrase recognition via dissimilarity significance classification. In *Proceedings of EMNLP*.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of EMNLP*.

Robbins, H. (1985). Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*. Springer.

Rus, V., McCarthy, P. M., Lintean, M. C., McNamara, D. S., and Graesser, A. C. (2008). Paraphrase identification with lexico-syntactic graph subsumption. In *Proceedings of FLAIRS*.

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of NIPS*.

Tran-Thanh, L., Stein, S., Rogers, A., and Jennings, N. R. (2012). Efficient crowdsourcing of unknown experts using multi-armed bandits. In *Proceedings of ECAI*.

Vanderwende, L., Suzuki, H., Brockett, C., and Nenkova, A. (2007). Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43.

Wan, S., Dras, M., Dale, R., and Paris, C. (2006). Using dependency-based features to take the para-farce out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop*.

Wang, L., Dyer, C., Black, A. W., and Trancoso, I. (2013). Paraphrasing 4 microblog normalization. In *Proceedings of EMNLP*.

Xu, W. (2014). *Data-Drive Approaches for Paraphrasing Across Language Variations*. PhD thesis, Department of Computer Science, New York University.

Xu, W., Ritter, A., Callison-Burch, C., Dolan, W. B., and Ji, Y. (2014). Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics (TACL)*, 2(1).

Xu, W., Ritter, A., and Grishman, R. (2013). Gathering and generating paraphrases from twitter with application to normalization. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*.

Zanzotto, F. M., Pennacchiotti, M., and Tsioutsiouliklis, K. (2011). Linguistic redundancy in twitter. In *Proceedings of EMNLP*.