Extracting Lexically Divergent Paraphrases from Twitter

Wei Xu[1], Alan Ritter[2], Chris Callison-Burch[1], William B. Dolan[3] and Yangfeng Ji[4]
[1] University of Pennsylvania, Philadelphia, PA, USA
{xwe, ccb}@cis.upenn.edu
[2] The Ohio State University, Columbus, OH, USA
ritter.1492@osu.edu
[3] Microsoft Research, Redmond, WA, USA
billdol@microsoft.com
[4] Georgia Institute of Technology, Atlanta, GA, USA
jiyfeng@gatech.edu

Abstract

We present MULTIP (Multi-instance Learning Paraphrase Model), a new model suited to identify paraphrases within the short messages on Twitter. We jointly model paraphrase relations between word and sentence pairs and assume only sentence-level annotations during learning. Using this principled latent variable model alone, we achieve the performance competitive with a state-of-the-art method which combines a latent space model with a feature-based supervised classifier. Our model also captures lexically divergent paraphrases that differ from yet complement previous methods; combining our model with previous work significantly outperforms the state-of-the-art. In addition, we present a novel annotation methodology that has allowed us to crowdsource a paraphrase corpus from Twitter. We make this new dataset available to the research community.

1 Introduction

Paraphrases are alternative linguistic expressions of the same or similar meaning (Bhagat and Hovy, 2013). Twitter engages millions of users, who naturally talk about the same topics simultaneously and frequently convey similar meaning using diverse linguistic expressions. The unique characteristics of this user-generated text presents new challenges and opportunities for paraphrase research (Xu et al., 2013b; Wang et al., 2013). For many applications, like automatic summarization, first story detection (Petrović et al., 2012) and search (Zanzotto et al., 2011), it is crucial to resolve redundancy in tweets (e.g. *oscar nom'd doc* ↔ *Oscar-nominated documentary*).

In this paper, we investigate the task of determining whether two tweets are paraphrases. Previous work has exploited a pair of shared named entities to locate semantically equivalent patterns from related news articles (Shinyama et al., 2002; Sekine, 2005; Zhang and Weld, 2013). But short sentences in Twitter do not often mention two named entities (Ritter et al., 2012) and require nontrivial generalization from named entities to other words. For example, consider the following two sentences about basketball player *Brook Lopez* from Twitter:

- ○ *That boy* ***Brook Lopez*** *with a deep* <u>*3*</u>
- ○ ***brook lopez*** *hit a* <u>*3*</u> *and i missed it*

Although these sentences do not have many words in common, the identical word *"3"* is a strong indicator that the two sentences are paraphrases.

We therefore propose a novel joint word-sentence approach, incorporating a multi-instance learning assumption (Dietterich et al., 1997) that two sentences under the same **topic** (we highlight topics in bold) are paraphrases if they contain at least one word pair (we call it an <u>anchor</u> and highlight with underscores; the words in the anchor pair need not be identical) that is indicative of sentential paraphrase. This *at-least-one-anchor* assumption might be ineffective for long or randomly paired sentences, but holds up better for short sentences that are temporally and topically related on Twitter. Moreover, our model design (see Figure 1) allows exploitation of arbitrary features and linguistic resources, such as part-of-speech features and a normalization lex-

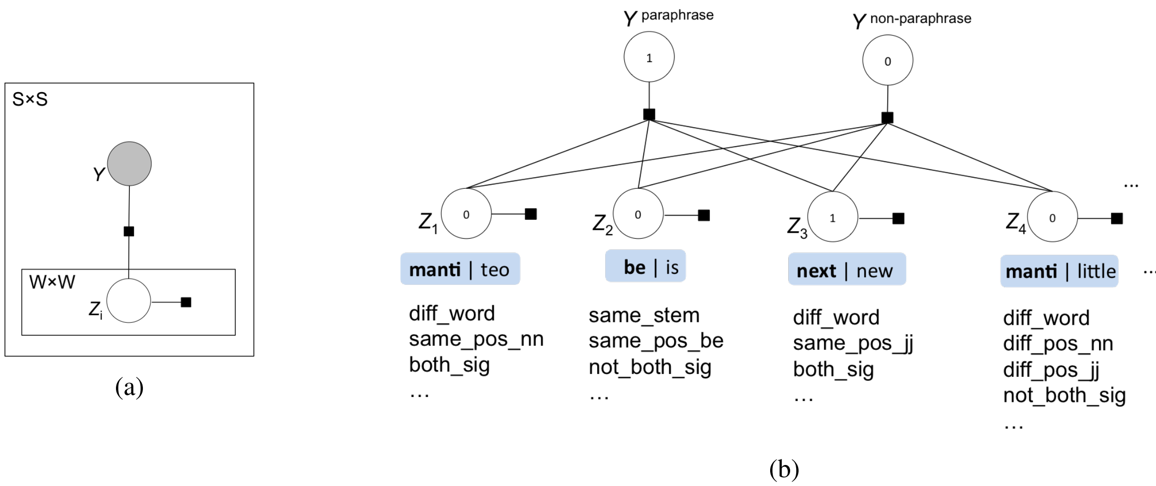

Figure 1: (a) a plate representation of the MULTIP model (b) an example instantiation of MULTIP for the pair of sentences "*Manti bout to be the* <u>*next*</u> ***Junior Seau***" and "*Teo is the little* <u>*new*</u> ***Junior Seau***", in which a new American football player *Manti Te'o* was being compared to a famous former player *Junior Seau*. Only 4 out of the total 6×5 word pairs, z_1 - z_{30}, are shown here.

icon, to discriminatively determine word pairs as paraphrastic anchors or not.

Our graphical model is a major departure from popular surface- or latent- similarity methods (Wan et al., 2006; Guo and Diab, 2012; Ji and Eisenstein, 2013, and others). Our approach to extract paraphrases from Twitter is general and can be combined with various topic detecting solutions. As a demonstration, we use Twitter's own trending topic service[1] to collect data and conduct experiments. While having a principled and extensible design, our model alone achieves performance on par with a state-of-the-art ensemble approach that involves both latent semantic modeling and supervised classification. The proposed model also captures radically different paraphrases from previous approaches; a combined system shows significant improvement over the state-of-the-art.

This paper makes the following contributions:

1) We present a novel latent variable model for paraphrase identification, that specifically accommodates the very short context and divergent wording in Twitter data. We experimentally compare several representative approaches and show that our proposed method yields state-of-the-art results and identifies paraphrases that are complementary to previous methods.

2) We develop an efficient crowdsourcing method and construct a Twitter Paraphrase Corpus of about 18,000 sentence pairs, as a first common testbed for the development and comparison of paraphrase identification and semantic similarity systems. We make this dataset available to the research community.[2]

2 Joint Word-Sentence Paraphrase Model

We present a new latent variable model that jointly captures paraphrase relations between sentence pairs and word pairs. It is very different from previous approaches in that its primary design goal and motivation is targeted towards short, lexically diverse text on the social web.

2.1 At-least-one-anchor Assumption

Much previous work on paraphrase identification has been developed and evaluated on a specific benchmark dataset, the Microsoft Research Paraphrase Corpus (Dolan et al., 2004), which is de-

[1]More information about Twitter's trends: `https://support.twitter.com/articles/101125-faqs-about-twitter-s-trends`

[2]The dataset and code are made available at: SemEval-2015 shared task `http://alt.qcri.org/semeval2015/task1/` and `https://github.com/cocoxu/twitterparaphrase/`

Corpus	Examples
News (Dolan and Brockett, 2005)	○ Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier. ○ With the scandal hanging over Stewart's company, revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.
	○ The Senate Select Committee on Intelligence is preparing a blistering report on prewar intelligence on Iraq. ○ American intelligence leading up to the war on Iraq will be criticized by a powerful US Congressional committee due to report soon, officials said today.
Twitter (This Work)	○ Can Klay Thompson wake up ○ Cmon Klay need u to get it going
	○ Ezekiel Ansah wearing 3D glasses wout the lens ○ Wait Ezekiel ansah is wearing 3d movie glasses with the lenses knocked out
	○ Marriage equality law passed in Rhode Island ○ Congrats to Rhode Island becoming the 10th state to enact marriage equality

Table 1: Representative examples from paraphrase corpora. The average sentence length is 11.9 words in Twitter vs. 18.6 in the news corpus.

rived from news articles. Twitter data is very different, as shown in Table 1. We observe that among tweets posted around the same time about the same topic (e.g. a named entity), sentential paraphrases are short and can often be *"anchored"* by lexical paraphrases. This intuition leads to the *at-least-one-anchor* assumption we stated in the introduction.

The anchor could be a word the two sentences share in common. It also could be a pair of different words. For example, the word pair *"next* $\|$ *new"* in two tweets about a new player *Manti Te'o* to a famous former American football player *Junior Seau*:

○ *Manti bout to be the <u>next</u>* ***Junior Seau***

○ *Teo is the little <u>new</u>* ***Junior Seau***

Further note that not every word pair of similar meaning indicates sentence-level paraphrase. For example, the word *"3"*, shared by two sentences about movie *"Iron Man"* that refers to the 3rd sequel of the movie, is **not** a paraphrastic anchor:

○ ***Iron Man*** *<u>3</u> was brilliant fun*

○ ***Iron Man*** *<u>3</u> tonight see what this is like*

Therefore, we use a discriminative model at the word-level to incorporate various features, such as part-of-speech features, to determine how probable a word pair is a paraphrase anchor.

2.2 Multi-instance Learning Paraphrase Model (MULTIP)

The *at-least-one-anchor* assumption naturally leads to a multi-instance learning problem (Dietterich et al., 1997), where the learner only observes labels on bags of instances (i.e. sentence-level paraphrases in this case) instead of labels on each individual instance (i.e. word pair).

We formally define an undirected graphical model of multi-instance learning for paraphrase identification – MULTIP. Figure 1 shows the proposed model in plate form and gives an example instantiation. The model has two layers, which allows joint reasoning between sentence-level and word-level components.

For each pair of sentences $s_i = (s_{i_1}, s_{i_2})$, there is *an aggregate binary variable* y_i that represents whether they are paraphrases, and which is observed in the labeled training data. Let $W(s_{i_k})$ be the set of words in the sentence s_{i_k}, excluding the topic names. For each word pair $w_j = (w_{j_1}, w_{j_2}) \in W(s_{i_1}) \times W(s_{i_2})$, there exists *a latent variable* z_j which denotes whether the word pair is a paraphrase anchor. In total there are $m = |W(s_{i_1})| \times |W(s_{i_2})|$ word pairs, and thus $\mathbf{z}_i = z_1, z_2, ..., z_j, ..., z_m$. Our *at-least-one-anchor* assumption is realized by a deterministic-or function; that is, if there exists at least one j such that $z_j = 1$, then the sentence pair is a paraphrase.

Our conditional paraphrase identification model is defined as follows:

$$
\begin{aligned}
P(\mathbf{z}_i, y_i | \mathbf{w}_i; \theta) &= \prod_{j=1}^{m} \phi(z_j, w_j; \theta) \times \sigma(\mathbf{z}_i, y_i) \\
&= \prod_{j=1}^{m} \exp(\theta \cdot f(z_j, w_j)) \times \sigma(\mathbf{z}_i, y_i)
\end{aligned} \tag{1}
$$

where $f(z_j, w_j)$ is a vector of features extracted for the word pair w_j, θ is the parameter vector, and σ is the factor that corresponds to the deterministic-or constraint:

$$
\sigma(\mathbf{z}_i, y_i) = \begin{cases} 1 & \text{if } y_i = true \wedge \exists j : z_j = 1 \\ 1 & \text{if } y_i = false \wedge \forall j : z_j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}
$$

2.3 Learning

To learn the parameters of the word-level paraphrase anchor classifier, θ, we maximize likelihood over the sentence-level annotations in our paraphrase corpus:

$$
\begin{aligned}
\theta^* &= \arg\max_{\theta} P(\mathbf{y}|\mathbf{w}; \theta) \\
&= \arg\max_{\theta} \prod_i \sum_{\mathbf{z}_i} P(\mathbf{z}_i, y_i | \mathbf{w}_i; \theta)
\end{aligned} \tag{3}
$$

An iterative gradient-ascent approach is used to estimate θ using perceptron-style additive updates (Collins, 2002; Liang et al., 2006; Zettlemoyer and Collins, 2007; Hoffmann et al., 2011). We define an update based on the gradient of the conditional log likelihood using Viterbi approximation, as follows:

$$
\begin{aligned}
\frac{\partial \log P(\mathbf{y}|\mathbf{w}; \theta)}{\partial \theta} &= \mathbf{E}_{P(\mathbf{z}|\mathbf{w},\mathbf{y};\theta)}\Big(\sum_i f(\mathbf{z}_i, \mathbf{w}_i)\Big) \\
&\quad - \mathbf{E}_{P(\mathbf{z},\mathbf{y}|\mathbf{w};\theta)}\Big(\sum_i f(\mathbf{z}_i, \mathbf{w}_i)\Big) \\
&\approx \sum_i f(\mathbf{z}_i^*, \mathbf{w}_i) - \sum_i f(\mathbf{z}_i', \mathbf{w}_i)
\end{aligned} \tag{4}
$$

where we define the feature sum for each sentence $f(\mathbf{z}_i, \mathbf{w}_i) = \sum_j f(z_j, w_j)$ over all word pairs.

These two above expectations are approximated by solving two simple inference problems as maximizations:

$$
\begin{aligned}
\mathbf{z}^* &= \arg\max_{\mathbf{z}} P(\mathbf{z}|\mathbf{w}, \mathbf{y}; \theta) \\
\mathbf{y}', \mathbf{z}' &= \arg\max_{\mathbf{y},\mathbf{z}} P(\mathbf{z}, \mathbf{y}|\mathbf{w}; \theta)
\end{aligned} \tag{5}
$$

Input: a training set $\{(\mathbf{s}_i, y_i) | i = 1...n\}$, where i is an index corresponding to a particular sentence pair $\mathbf{s}_i$, and y_i is the training label.

initialize parameter vector $\theta \leftarrow \mathbf{0}$
for $i \leftarrow 1$ **to** n **do**
  extract all possible word pairs $\mathbf{w}_i = w_1, w_2, ..., w_m$ and their features from the sentence pair $\mathbf{s}_i$
end for
for $l \leftarrow 1$ **to** maximum iterations **do**
  for $i \leftarrow 1$ **to** n **do**
    $(y_i', \mathbf{z}_i') \leftarrow \arg\max_{y_i, \mathbf{z}_i} P(\mathbf{z}_i, y_i | \mathbf{w}_i; \theta)$
    if $y_i' \neq y_i$ **then**
      $\mathbf{z}_i^* \leftarrow \arg\max_{\mathbf{z}_i} P(\mathbf{z}_i | \mathbf{w}_i, y_i; \theta)$
      $\theta \leftarrow \theta + f(\mathbf{z}_i^*, \mathbf{w}_i) - f(\mathbf{z}_i', \mathbf{w}_i)$
    end if
  end for
end for
return model parameters θ

Figure 2: MULTIP Learning Algorithm

Computing both $\mathbf{z}'$ and $\mathbf{z}^*$ are rather straightforward under the structure of our model and can be solved in time linear in the number of word pairs. The dependencies between $\mathbf{z}$ and $\mathbf{y}$ are defined as deterministic-or factors $\sigma(\mathbf{z}_i, y_i)$, which when satisfied do not affect the overall probability of the solution. Each sentence pair is independent conditioned on the parameters. For $\mathbf{z}'$, it is sufficient to independently compute the most likely assignment $\mathbf{z}_i'$ for each word pair, ignoring the deterministic dependencies. $\mathbf{y}_i'$ is then set by aggregating all $\mathbf{z}_i'$ through the deterministic-or operation. Similarly, we can find the exact solution for $\mathbf{z}^*$, the most likely assignment that respects the sentence-level training label $\mathbf{y}$. For a positive training instance, we simply find its highest scored word pair w_τ by the word-level classifier, then set $z_\tau^* = 1$ and $z_j^* = \arg\max_{x \in 0,1} \phi(x, w_j; \theta)$ for all $j \neq \tau$; for a negative example, we set $\mathbf{z}_i^* = \mathbf{0}$. The time complexity of both inferences for one sentence pair is $O(|W(s)|^2)$, where $|W(s)|^2$ is the number of word pairs.

In practice, we use online learning instead of optimizing the full objective. The detailed learning algorithm is presented in Figure 2. Following Hoffmann et al. (2011), we use 50 iterations in the experiments.

2.4 Feature Design

At the word-level, our discriminative model allows use of arbitrary features that are similar to those in monolingual word alignment models (MacCartney et al., 2008; Thadani and McKeown, 2011; Yao et al., 2013a,b). But unlike discriminative monolingual word alignment, we only use sentence-level training labels instead of word-level alignment annotation. For every word pair, we extract the following features:

String Features that indicate whether the two words, their stemmed forms and their normalized forms are the same, similar or dissimilar. We used the Morpha stemmer (Minnen et al., 2001),[3] Jaro-Winkler string similarity (Winkler, 1999) and the Twitter normalization lexicon by Han et al. (2012).

POS Features that are based on the part-of-speech tags of the two words in the pair, specifying whether the two words have same or different POS tags and what the specific tags are. We use the Twitter Part-Of-Speech tagger developed by Derczynski et al. (2013). We add new fine-grained tags for variations of the eight words: "a", "be", "do", "have", "get", "go", "follow" and "please". For example, we use a tag *HA* for words "have", "has" and "had".

Topical Features that relate to the strength of a word's association to the topic. This feature identifies the popular words in each topic, e.g. *"3"* in tweets about basketball game, *"RIP"* in tweets about a celebrity's death. We use G^2 log-likelihood-ratio statistic, which has been frequently used in NLP, as a measure of word associations (Dunning, 1993; Moore, 2004). The significant scores are computed for each trend on an average of about 1500 sentences and converted to binary features for every word pair, indicating whether the two words are both significant or not.

Our topical features are novel and were not used in previous work. Following Riedel et al. (2010) and Hoffmann et al. (2011), we also incorporate conjunction features into our system for better accuracy, namely Word+POS, Word+Topical and Word+POS+Topical features.

[3] `https://github.com/knowitall/morpha`

3 Experiments

3.1 Data

It is nontrivial to gather a gold-standard dataset of naturally occurring paraphrases and non-paraphrases efficiently from Twitter, since this requires pairwise comparison of tweets and faces a very large search space. To make this annotation task tractable, we design a novel and efficient crowdsourcing method using Amazon Mechanical Turk. Our entire data collection process is detailed in Section §4, with several experiments that demonstrate annotation quality and efficiency.

In total, we constructed a Twitter Paraphrase Corpus of 18,762 sentence pairs and 19,946 unique sentences. The training and development set consists of 17,790 sentence pairs posted between April 24th and May 3rd, 2014 from 500+ trending topics (excluding hashtags). Our paraphrase model and data collection approach is general and can be combined with various Twitter topic detecting solutions (Diao et al., 2012; Ritter et al., 2012). As a demonstration, we use Twitter's own trends service since it is easily available. Twitter trending topics are determined by an unpublished algorithm, which finds words, phrases and hashtags that have had a sharp increase in popularity, as opposed to overall volume. We use case-insensitive exact matching to locate topic names in the sentences.

Each sentence pair was annotated by 5 different crowdsourcing workers. For the test set, we obtained both crowdsourced and expert labels on 972 sentence pairs from 20 randomly sampled Twitter trending topics between May 13th and June 10th. Our dataset is more realistic and balanced, containing 79% non-paraphrases vs. 34% in the benchmark Microsoft Paraphrase Corpus of news data. As noted in (Das and Smith, 2009), the lack of natural non-paraphrases in the MSR corpus creates bias towards certain models.

3.2 Baselines

We use four baselines to compare with our proposed approach for the sentential paraphrase identification task. For the first baseline, we choose 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

Method	F1	Precision	Recall
Random	0.294	0.208	0.500
WTMF (Guo and Diab, 2012)*	0.583	0.525	0.655
LR (Das and Smith, 2009)**	0.630	0.629	0.632
LEXLATENT	0.641	0.663	0.621
LEXDISCRIM (Ji and Eisenstein, 2013)	0.645	0.664	0.628
MULTIP	0.724	0.722	0.726
Human Upperbound	0.823	0.752	0.908

Table 2: Performance of different paraphrase identification approaches on Twitter data. *An enhanced version that uses additional 1.6 million sentences from Twitter. ** Reimplementation of a strong baseline used by Das and Smith (2009).

competitive performance on the MSR corpus.

The second baseline is a state-of-the-art unsupervised method, Weighted Textual Matrix Factorization (WTMF),[4] which 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 original model was learned from WordNet (Fellbaum, 2010), OntoNotes (Hovy et al., 2006), Wiktionary, the Brown corpus (Francis and Kucera, 1979). We enhance the model with 1.6 million sentences from Twitter as suggested by Guo et al. (2013).

Ji and Eisenstein (2013) presented a state-of-the-art ensemble system, which we call LEXDISCRIM.[5] It directly combines both discriminatively-tuned latent features and surface lexical features into a SVM classifier. Specifically, the latent representation of a pair of sentences $\vec{v_1}$ and $\vec{v_2}$ is converted into a feature vector, $[\vec{v_1}+\vec{v_2}, |\vec{v_1}-\vec{v_2}|]$, by concatenating the element-wise sum $\vec{v_1}+\vec{v_2}$ and absolute different $|\vec{v_1}-\vec{v_2}|$.

We also introduce a new baseline, LEXLATENT, which is a simplified version of LEXDISCRIM and easy to reproduce. It uses the same method to combine latent features and surface features, but combines the open-sourced WTMF latent space model and the logistic regression model from above instead. It achieves similar performance as LEXDISCRIM on our dataset (Table 2).

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

[5]The parsing feature was removed because it was not helpful on our Twitter dataset.

3.3 System Performance

For evaluation of different systems, we compute precision-recall curves and report the highest F1 measure of any point on the curve, on the test dataset of 972 sentence pairs against the expert labels. Table 2 shows the performance of different systems. Our proposed MULTIP, a principled latent variable model alone, achieves competitive results with the state-of-the-art system that combines discriminative training and latent semantics.

In Table 2, we also show the agreement levels of labels derived from 5 non-expert annotations on Mechanical Turk, which can be considered as an upperbound for automatic paraphrase recognition task performed on this data set. The annotation quality of our corpus is surprisingly good given the fact that the definition of paraphrase is rather inexact (Bhagat and Hovy, 2013); the inter-rater agreement between expert annotators on news data is only 0.83 as reported by Dolan et al. (2004).

	F1	Prec	Recall
MULTIP	0.724	0.722	0.726
- String features	0.509	0.448	0.589
- POS features	0.496	0.350	0.851
- Topical features	0.715	0.694	0.737

Table 3: Feature ablation by removing each individual feature group from the full set.

To assess the impact of different features on the model's performance, we conduct feature ablation experiments, removing one group of features at a time. The results are shown in Table 3. Both string

Para?	Sentence Pair from Twitter	MULTIP	LEXLATENT
YES	∘ The new **Ciroc** flavor has arrived ∘ **Ciroc** got a new flavor comin out	rank=12	rank=266
YES	∘ **Roberto Mancini** gets the boot from Man City ∘ **Roberto Mancini** has been sacked by Manchester City with the Blues saying	rank=64	rank=452
YES	∘ I want to watch **the purge** tonight ∘ I want to go see **The Purge** who wants to come with	rank=136	rank=11
NO	∘ Somebody took the **Marlins** to 20 innings ∘ Anyone who stayed 20 innings for the **marlins**	rank= 8	rank=54
NO	∘ **WORLD OF JENKS** IS ON AT 11 ∘ **World of Jenks** is my favorite show on tv	rank=167	rank=9

Table 4: Example system outputs; *rank* is the position in the list of all candidate paraphrase pairs in the test set ordered by model score. MULTIP discovers lexically divergent paraphrases while LEXLATENT prefers more overall sentence similarity. Underline marks the word pair(s) with highest estimated probability as paraphrastic anchor(s) for each sentence pair.

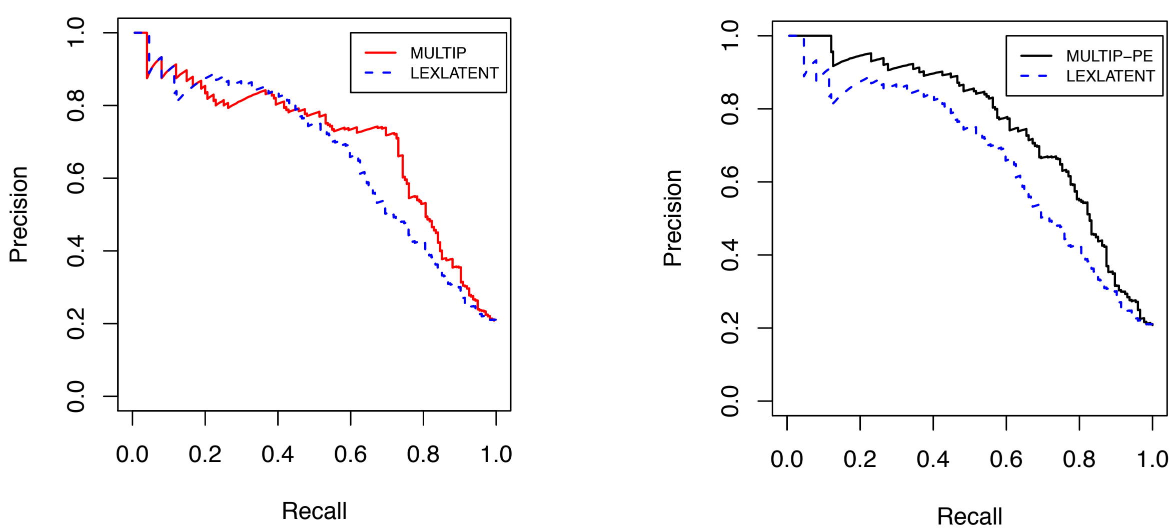

Figure 3: Precision and recall curves. Our MULTIP model alone achieves competitive performance with the LEXLATENT system that combines latent space model and feature-based supervised classifier. The two approaches have complementary strengths, and achieves significant improvement when combined together (MULTIP-PE).

and POS features are essential for system performance, while topical features are helpful but not as crucial.

Figure 3 presents precision-recall curves and shows the sensitivity and specificity of each model in comparison. In the first half of the curve (recall < 0.5), MULTIP model makes bolder and less accurate decisions than LEXLATENT. However, the curve for MULTIP model is more flat and shows consistently better precision at the second half (recall > 0.5) as well as a higher maximum F1 score. This result reflects our design concept of MULTIP, which is intended to pick up sentential paraphrases with more divergent wordings aggressively. LEXLATENT, as a combined system, considers sentence features in both surface and latent space and is more conservative. Table 4 further illustrates this difference with some example system outputs.

3.4 Product of Experts (MULTIP-PE)

Our MULTIP model and previous similarity-based approaches have complementary strengths, so we experiment with combining MULTIP (P_m) and LEXLATENT (P_l) through a product of experts (Hinton, 2002):

$$P(\mathbf{y}|s_1, s_2) = \frac{P_m(\mathbf{y}|s_1, s_2) \times P_l(\mathbf{y}|s_1, s_2)}{\sum_{\mathbf{y}} P_m(\mathbf{y}|s_1, s_2) \times P_l(\mathbf{y}|s_1, s_2)} \quad (6)$$

The resulting system MULTIP-PE provides consistently better precision and recall over the LEXLATENT model, as shown on the right in Figure 3. The MULTIP-PE system outperforms LEXLATENT significantly according to a paired t-test with ρ less than 0.05. Our proposed MULTIP takes advantage of Twitter's specific properties and provides complementary information to previous approaches. Previously, Das and Smith (2009) has also used a product of experts to combine a lexical and a syntax-based model together.

4 Constructing Twitter Paraphrase Corpus

We now turn to describing our data collection and annotation methodology. Our goal is to construct a high quality dataset that contains representative examples of paraphrases and non-paraphrases in Twitter. 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 (less than 8%) of expressing the same meaning. 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.[6] 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 and split each tweet into sentences.[7]

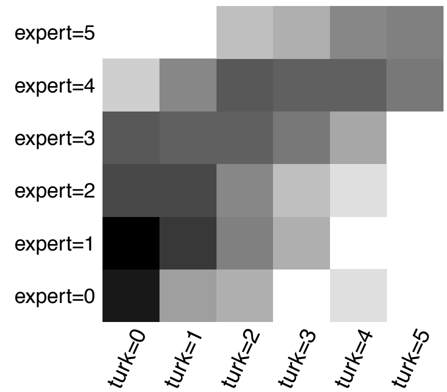

Figure 4: A heat-map showing overlap between expert and crowdsourcing annotation. The intensity along the diagonal indicates good reliability of crowdsourcing workers for this particular task; and the shift above the diagonal reflects the difference between the two annotation schemas. For crowdsourcing (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 are discarded in training and testing in our experiments.

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 Poe-

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

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

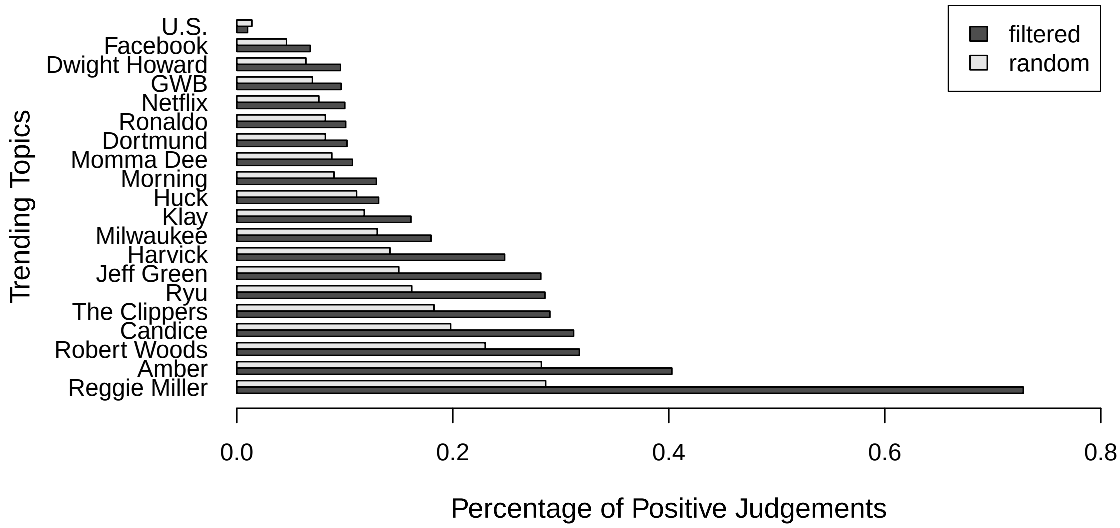

Figure 5: The proportion of paraphrases (percentage of positive votes from annotators) vary greatly across different topics. Automatic filtering in Section 4.4 roughly doubles the paraphrase yield.

sio, 2008) with other annotators. We also compute inter-annotator agreement with an expert annotator on 971 sentence pairs. In the expert annotation, we adopt a 5-point Likert scale to measure the degree of semantic similarity between 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 4 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.

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.

$$Salience(s) = \sum_{w_i \in s} \frac{P(w_i)}{|\{w_i | w_i \in s\}|} \tag{7}$$

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 5 and Figure 6). We also use PINC (Chen and Dolan, 2011) to measure the quality of paraphrases collected (Figure 7). 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.

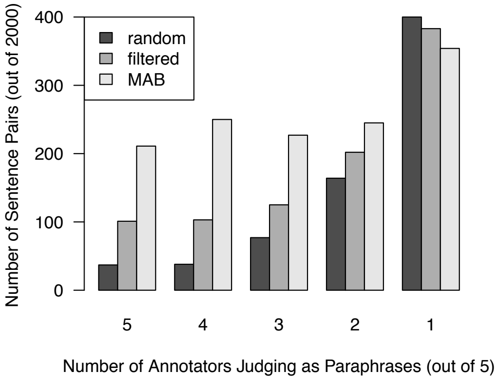

Figure 6: Numbers of paraphrases collected by different methods. The annotation efficiency (3,4,5 are regarded as paraphrases) is significantly improved by the sentence filtering and Multi-Armed Bandits (MAB) based topic selection.

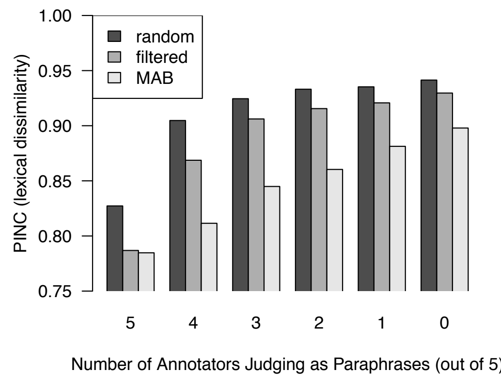

Figure 7: PINC scores of paraphrases collected. The higher the PINC, the more significant the rewording. Our proposed annotation strategy quadruples paraphrase yield, while not greatly reducing diversity as measured by PINC.

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 5). 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 6).

5 Related Work

Automatic Paraphrase Identification has been widely studied (Androutsopoulos and Malakasiotis, 2010; Madnani and Dorr, 2010). The ACL Wiki gives an excellent summary of various techniques.[8] Many recent high-performance approaches use system combination (Das and Smith, 2009; Madnani et al., 2012; Ji and Eisenstein, 2013). For example, Madnani et al. (2012) combines multiple sophisticated machine translation metrics using a meta-classifier. An earlier attempt on Twitter data is that of Xu et al. (2013b). They limited the search space to only the tweets that explicitly mention a same date and a same named entity, however there remain a considerable amount of mislabels in their data.[9] Zanzotto et al. (2011) also experimented with SVM tree kernel methods on Twitter data.

Departing from the previous work, we propose a latent variable model to jointly infer the correspondence between words and sentences. It is related to discriminative monolingual word alignment (MacCartney et al., 2008; Thadani and McKeown,

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

[9] The data is released by Xu et al. (2013b) at: `https://github.com/cocoxu/twitterparaphrase/`

2011; Yao et al., 2013a,b), but different in that the paraphrase task requires additional sentence alignment modeling with no word alignment data. Our approach is also inspired by Fung and Cheung's (2004a; 2004b) work on bootstrapping bilingual parallel sentence and word translations from comparable corpora.

Multiple Instance Learning (Dietterich et al., 1997) has been used by different research groups in the field of information extraction (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Ritter et al., 2013; Xu et al., 2013a). The idea is to leverage structured data as weak supervision for tasks such as relation extraction. This is done, for example, by making the assumption that at least one sentence in the corpus which mentions a pair of entities (e_1, e_2) participating in a relation (r) expresses the proposition: $r(e_1, e_2)$.

Crowdsourcing Paraphrase Acquisition: Buzek et al. (2010) and Denkowski et al. (2010) focused specifically on collecting paraphrases of text to be translated to improve machine translation quality. Chen and Dolan (2011) gathered a large-scale paraphrase corpus by asking Mechanical Turk workers to caption the action in short video segments. Similarly, Burrows et al. (2012) asked crowdsourcing workers to rewrite selected excerpts from books. Ling et al. (2014) crowdsourced bilingual parallel text using Twitter as the source of data.

In contrast, we design a simple crowdsourcing task requiring only binary judgements on sentences collected from Twitter. There are several advantages as compared to existing work: a) the corpus also covers a very diverse range of topics and linguistic expressions, especially colloquial language, which is different from and thus complements previous paraphrase corpora; b) the paraphrase corpus collected contains a representative proportion of both negative and positive instances, while lack of good negative examples was an issue in the previous research (Das and Smith, 2009); c) this method is scalable and sustainable due to the simplicity of the task and real-time, virtually unlimited text supply from Twitter.

6 Conclusions

This paper introduced MULTIP, a joint word-sentence model to learn paraphrases from temporally and topically grouped messages in Twitter. While simple and principled, our model achieves performance competitive with a state-of-the-art ensemble system combining latent semantic representations and surface similarity. By combining our method with previous work as a product-of-experts we outperform the state-of-the-art. Our latent-variable approach is capable of learning word-level paraphrase anchors given only sentence annotations. Because our graphical model is modular and extensible (for example it should be possible to replace the deterministic-or with other aggregators), we are optimistic this work might provide a path towards weakly supervised word alignment models using only sentence-level annotations.

In addition, we presented a novel and efficient annotation methodology which was used to crowdsource a unique corpus of paraphrases harvested from Twitter. We make this resource available to the research community.

Acknowledgments

The author would like to thank editor Sharon Goldwater and three anonymous reviewers for their thoughtful comments, which substantially improved this paper. We also thank Ralph Grishman, Sameer Singh, Yoav Artzi, Mark Yatskar, Chris Quirk, Ani Nenkova and Mitch Marcus for their feedback.

This material is based in part on research sponsored by the NSF under grant IIS-1430651, DARPA under agreement number FA8750-13-2-0017 (the DEFT program) and through a Google Faculty Research Award to Chris Callison-Burch. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government. Yangfeng Ji is supported by a Google Faculty Research Award awarded to Jacob Eisenstein.

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 the First Joint Conference on Lexical and Computational Semantics (*SEM)*.

Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38.

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

Bhagat, R. and Hovy, E. (2013). What is a paraphrase? *Computational Linguistics*, 39(3).

Burrows, S., Potthast, M., and Stein, B. (2012). Paraphrase acquisition via crowdsourcing and machine learning. *Transactions on Intelligent Systems and Technology (ACM TIST)*.

Buzek, O., Resnik, P., and Bederson, B. B. (2010). Error driven paraphrase annotation using Mechanical Turk. In *Proceedings of the Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Das, D. and Smith, N. A. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*.

Denkowski, M., Al-Haj, H., and Lavie, A. (2010). Turker-assisted paraphrasing for English-Arabic machine translation. In *Proceedings of the Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

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 the Recent Advances in Natural Language Processing (RANLP)*.

Diao, Q., Jiang, J., Zhu, F., and Lim, E.-P. (2012). Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1).

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*.

Dolan, W. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the 3rd International Workshop on Paraphrasing*.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1).

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

Francis, W. N. and Kucera, H. (1979). *Brown corpus manual*. Brown University.

Fung, P. and Cheung, P. (2004a). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Fung, P. and Cheung, P. (2004b). Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Guo, W. and Diab, M. (2012). Modeling sentences in the latent space. In *Proceedings of the 50th*

Annual Meeting of the Association for Computational Linguistics (ACL).

Guo, W., Li, H., Ji, H., and Diab, M. (2013). Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Han, B., Cook, P., and Baldwin, T. (2012). Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8).

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L. S., and Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL)*.

Ji, Y. and Eisenstein, J. (2013). Discriminative improvements to distributional sentence similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Liang, P., Bouchard-Côté, A., Klein, D., and Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (COLING-ACL)*.

Ling, W., Marujo, L., Dyer, C., Alan, B., and Isabel, T. (2014). Crowdsourcing high-quality parallel data extraction from Twitter. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT)*.

MacCartney, B., Galley, M., and Manning, C. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Madnani, N. and Dorr, B. J. (2010). Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3).

Madnani, N., Tetreault, J., and Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*.

Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of english. *Natural Language Engineering*, 7(03).

Moore, R. C. (2004). On log-likelihood-ratios and the significance of rare events. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Petrović, S., Osborne, M., and Lavrenko, V. (2012). Using paraphrases for improving first story detection in news and Twitter. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedigns of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.

Ritter, A., Mausam, Etzioni, O., and Clark, S. (2012). Open domain event extraction from Twitter. In *Proceedings of the 18th International Con-*

ference on Knowledge Discovery and Data Mining (SIGKDD).

Ritter, A., Zettlemoyer, L., Mausam, and Etzioni, O. (2013). Modeling missing data in distant supervision for information extraction. *Transactions of the Association for Computational Linguistics (TACL)*.

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

Sekine, S. (2005). Automatic paraphrase discovery based on context and keywords between NE pairs. In *Proceedings of the 3rd International Workshop on Paraphrasing*.

Shinyama, Y., Sekine, S., and Sudo, K. (2002). Automatic paraphrase acquisition from news articles. In *Proceedings of the 2nd International Conference on Human Language Technology Research (HLT)*.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thadani, K. and McKeown, K. (2011). Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics - Human Language Technologies (ACL-HLT)*.

Tran-Thanh, L., Stein, S., Rogers, A., and Jennings, N. R. (2012). Efficient crowdsourcing of unknown experts using multi-armed bandits. In *Proceedings of the European Conference on Artificial Intelligence (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 the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Winkler, W. E. (1999). The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau.

Xu, W., Hoffmann, R., Zhao, L., and Grishman, R. (2013a). Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Yao, X., Van Durme, B., Callison-Burch, C., and Clark, P. (2013a). A lightweight and high performance monolingual word aligner. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yao, X., Van Durme, B., and Clark, P. (2013b). Semi-markov phrase-based monolingual alignment. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Zanzotto, F. M., Pennacchiotti, M., and Tsioutsiouliklis, K. (2011). Linguistic redundancy in Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zettlemoyer, L. S. and Collins, M. (2007). Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Zhang, C. and Weld, D. S. (2013). Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.