Extracting Lexically Divergent Paraphrases from Twitter

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

Abstract

We present \textsc{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 crowdsourced 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. \textit{oscar nom’d doc} $\leftrightarrow$ \textit{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:

- \textit{That boy Brook Lopez with a deep 3}
- \textit{brook lopez hit a 3 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 \textbf{topic} (we highlight topics in bold) are paraphrases if they contain at least one word pair (we call it an anchor and highlight with underscores; the words in the anchor pair need not be identical) that is indicative of sentential paraphrase. This \textit{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 MULTI P model (b) an example instantiation of MULTI P for the pair of sentences “Manti bout to be the next Junior Seau” and “Teo is the little new 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 $6 \times 5$ word pairs, $z_1 - z_{30}$, are shown here.

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\footnote{More information about Twitter’s trends: https://support.twitter.com/articles/101125-faqs-about-twitter-s-trends} 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.\footnote{The dataset and code are made available at: SemEval-2015 shared task http://alt.qcri.org/semeval2015/task1/ and https://github.com/cocoxu/twitterparaphrase/}

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-
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 without 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 next Junior Seau
 ◦ Teo is the little new 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 3 was brilliant fun
 ◦ Iron Man 3 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 $s_i = (s_{i1}, s_{i2})$, there is an aggregate binary variable $y_i$ that represents whether they are paraphrases, and which is observed in the labeled training data. Let $W(s_{ik})$ be the set of words in the sentence $s_{ik}$, excluding the topic names. For each word pair $w_j = (w_{j1}, w_{j2}) \in W(s_{i1}) \times W(s_{i2})$, there exists a latent variable $z_j$ which denotes whether the word pair is a paraphrase anchor. In total there are $m = |W(s_{i1})| \times |W(s_{i2})|$ word pairs, and thus $z_i = z_1, z_2, ..., z_j, ..., z_m$. 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:

\[
P(z_i, y_i|w_i; \theta) = \prod_{j=1}^{m} \phi(z_j, w_j; \theta) \times \sigma(z_i, y_i)
\]

\[
= \prod_{j=1}^{m} \exp(\theta \cdot f(z_j, w_j)) \times \sigma(z_i, y_i)
\]

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

\[
\sigma(z_i, y_i) = \begin{cases} 
1 & \text{if } y_i = \text{true} \land \exists j : z_j = 1 \\
1 & \text{if } y_i = \text{false} \land \forall j : z_j = 0 \\
0 & \text{otherwise}
\end{cases}
\]

**2.3 Learning**

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

\[
\theta^* = \arg \max_{\theta} P(y|w; \theta) = \arg \max_{\theta} \prod \sum P(z_i, y_i|w_i; \theta)
\]

An iterative gradient-ascent approach is used to estimate \( \theta \) 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:

\[
\frac{\partial \log P(y|w; \theta)}{\partial \theta} = E_{P(z|w,y;\theta)}\left( \sum_i f(z_i, w_i) \right) - E_{P(z,y|w;\theta)}\left( \sum_i f(z_i, w_i) \right) \\
\approx \sum_i f(z_i, w_i) - \sum_i f(z'_i, w_i)
\]

where we define the feature sum for each sentence \( f(z_i, w_i) = \sum_j f(z_j, w_j) \) over all word pairs.

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

\[
z^* = \arg \max_z P(z|w, y; \theta)
\]

\[
y', z' = \arg \max_{y,z} P(z, y|w; \theta)
\]

**Figure 2: MULTIP Learning Algorithm**

Computing both \( z' \) and \( z^* \) 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 \( z \) and \( y \) are defined as deterministic-or factors \( \sigma(z_i, y_i) \), which when satisfied do not affect the overall probability of the solution. Each sentence pair is independent conditioned on the parameters. For \( z' \), it is sufficient to independently compute the most likely assignment \( z'_i \) for each word pair, ignoring the deterministic dependencies. \( y'_i \) is then set by aggregating all \( z'_i \) through the deterministic-or operation. Similarly, we can find the exact solution for \( z^* \), the most likely assignment that respects the sentence-level training label \( y \). For a positive training instance, we simply find its highest scored word pair \( w \tau \) by the word-level classifier, then set \( z^* = 1 \) and \( y^* = \arg \max_{x \in 0,1} \phi(x, w_j; \theta) \) for all \( j \neq \tau \); for a negative example, we set \( z^* = 0 \). 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), 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 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

---

3 https://github.com/knowitall/morpha
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).

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.294</td>
<td>0.208</td>
<td>0.500</td>
</tr>
<tr>
<td>WTMF (Guo and Diab, 2012)*</td>
<td>0.583</td>
<td>0.525</td>
<td>0.655</td>
</tr>
<tr>
<td>LR (Das and Smith, 2009)**</td>
<td>0.630</td>
<td>0.629</td>
<td>0.632</td>
</tr>
<tr>
<td>LEXLATENT</td>
<td>0.641</td>
<td>0.663</td>
<td>0.621</td>
</tr>
<tr>
<td>LEXDISCRIM (Ji and Eisenstein, 2013)</td>
<td>0.645</td>
<td>0.664</td>
<td>0.628</td>
</tr>
<tr>
<td>MULTI P</td>
<td>0.724</td>
<td>0.722</td>
<td>0.726</td>
</tr>
<tr>
<td>Human Upperbound</td>
<td>0.823</td>
<td>0.752</td>
<td>0.908</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>F1</th>
<th>Prec</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTI P</td>
<td>0.724</td>
<td>0.722</td>
<td>0.726</td>
</tr>
<tr>
<td>- String features</td>
<td>0.509</td>
<td>0.448</td>
<td>0.589</td>
</tr>
<tr>
<td>- POS features</td>
<td>0.496</td>
<td>0.350</td>
<td>0.851</td>
</tr>
<tr>
<td>- Topical features</td>
<td>0.715</td>
<td>0.694</td>
<td>0.737</td>
</tr>
</tbody>
</table>

The second baseline is a state-of-the-art unsupervised method, Weighted Textual Matrix Factorization (WTMF), 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. 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).

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 MULTI P, 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).

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...
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.

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

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 (MULTI-P-PE)

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

$$P(y|s_1, s_2) = \frac{P_m(y|s_1, s_2) \times P_l(y|s_1, s_2)}{\sum_y P_m(y|s_1, s_2) \times P_l(y|s_1, s_2)}$$

(6)

The resulting system MULTI-P-PE provides consistently better precision and recall over the LEXLATENT model, as shown on the right in Figure 3. The MULTI-P-PE system outperforms LEXLATENT significantly according to a paired t-test with $\rho$ less than 0.05. Our proposed MULTI-P 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. 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.\footnote{More information about Twitter’s APIs: https://dev.twitter.com/docs/api/1.1/overview}

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 ∼ 45 seconds to answer.

4.3 Annotation Quality

We remove problematic annotators by checking their Cohen’s Kappa agreement (Artstein and Poe-
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.

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

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 naive 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.
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 $\epsilon$-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, $\epsilon$, 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 $\epsilon$ 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. 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 metaclassifier. 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.

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, 2010). 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.

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


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


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).


