Harvesting Parallel News Streams to Generate Paraphrases of Event Relations

Congle Zhang, Daniel S. Weld
Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
{clzhang,weld}@cs.washington.edu

Abstract

The distributional hypothesis, which states that words that occur in similar contexts tend to have similar meanings, has inspired several Web mining algorithms for paraphrasing semantically equivalent phrases. Unfortunately, these methods have several drawbacks, such as confusing synonyms with antonyms and causes with effects. This paper introduces three Temporal Correspondence Heuristics, that characterize regularities in parallel news streams, and shows how they may be used to generate high precision paraphrases for event relations. We encode the heuristics in a probabilistic graphical model to create the NEWSSPIKE algorithm for mining news streams. We present experiments demonstrating that NEWSSPIKE significantly outperforms several competitive baselines. In order to spur further research, we provide a large annotated corpus of timestamped news articles as well as the paraphrases produced by NEWSSPIKE.

1 Introduction

Paraphrasing, the task of finding sets of semantically equivalent surface forms, is crucial to many natural language processing applications, including relation extraction (Bhagat and Ravichandran, 2008), question answering (Fader et al., 2013), summarization (Barzilay et al., 1999) and machine translation (Callison-Burch et al., 2006). While the benefits of paraphrasing have been demonstrated, creating a large-scale corpus of high precision paraphrases remains a challenge — especially for event relations.

Many researchers have considered generating paraphrases by mining the Web guided by the *distributional hypothesis*, which states that words occurring in similar contexts tend to have similar meanings (Harris, 1954). For example, DIRT (Lin and Pantel, 2001) and Resolver (Yates and Etzioni, 2009) identify synonymous relation phrases by the distributions of their arguments. However, the distributional hypothesis has several drawbacks. First, it can confuse antonyms with synonyms because antonymous phrases appear in similar contexts as often as synonymous phrases. For the same reasons, it also often confuses causes with effects. For example, DIRT reports that the closest phrase to *fall* is *rise*, and the closest phrase to *shoot* is *kill*.[1] Second, the distributional hypothesis relies on statistics over large corpora to produce accurate similarity statistics. It remains unclear how to accurately paraphrase less frequent relations with the distributional hypothesis.

Another common approach employs the use of parallel corpora. News articles are an interesting target, because there often exist articles from different sources describing the same daily events. This peculiar property allows the use of the temporal assumption, which assumes that phrases in articles published at the same time tend to have similar meanings. For example, the approaches by Dolan *et al.* (2004) and Barzilay *et al.* (2003) identify pairs of sentential paraphrases in similar articles that have appeared in the same period of time. While these approaches use temporal information as a coarse filter in the data generation stage, they still largely rely on text metrics in the prediction stage. This not only reduces precision, but also limits the discovery of paraphrases with dissimilar sur-

[1] http://demo.patrickpantel.com/demos/lexsem/paraphrase.htm

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1776–1786,
Seattle, Washington, USA, 18-21 October 2013.

face strings.

The goal of our research is to develop a technique to generate paraphrases for large numbers of event relation with high precision, using only minimal human effort. The key to our approach is a joint cluster model using the temporal attributes of news streams, which allows us to identify semantic equivalence of event relation phrases with greater precision. In summary, this paper makes the following contributions:

- We formulate a set of three *temporal correspondence heuristics* that characterize regularities over parallel news streams.
- We develop a novel program, NEWSSPIKE, based on a probabilistic graphical model that jointly encodes these heuristics. We present inference and learning algorithms for our model.
- We present a series of detailed experiments demonstrating that NEWSSPIKE outperforms several competitive baselines, and show through ablation tests how each of the temporal heuristics affects performance.
- To spur further research on this topic, we provide both our generated paraphrase clusters and a corpus of 0.5M time-stamped news articles[2], collected over a period of about 50 days from hundreds of news sources.

2 System Overview

The main goal of this work is to generate high precision paraphrases for relation phrases. News streams are a promising resource, since articles from different sources tend to use semantically equivalent phrases to describe the same daily events. For example, when a recent scandal hit, headlines read: *"Armstrong steps down from Livestrong"*; *"Armstrong resigns from Livestrong"* and *"Armstrong cuts ties with Livestrong"*. From these we can conclude that the following relation phrases are semantically similar: {*step down from, resign from, cut ties with*}.

To realize this intuition, our first challenge is to represent an event. In practice, a question like *"What happened to Armstrong and Livestrong on Oct 17?"* could often lead to a unique answer. It implies that using an argument pair and a time-stamp could be an effective way to identify an event (*e.g. (Armstrong, Livestrong, Oct 17)* for the previous question). Based on this observation, this paper introduces a novel mechanism to paraphrase relations as summarized in Figure 1.

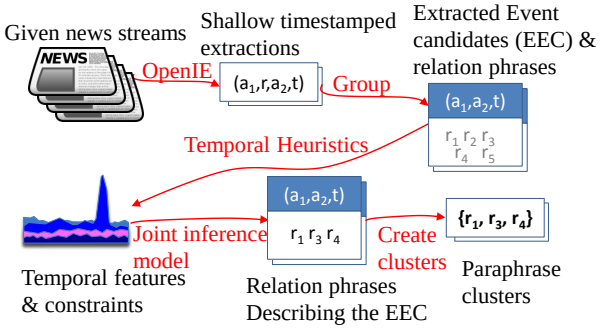

Figure 1: NEWSSPIKE first applies open information extraction to articles in the news streams, obtaining shallow extractions with time-stamps. Next, an *extracted event candidate* (EEC) is obtained after grouping daily extractions by argument pairs. Temporal features and constraints are developed based on our temporal correspondence heuristics and encoded into a joint inference model. The model finally creates the paraphrase clusters by predicting the relation phrases that describe the EEC.

NEWSSPIKE first applies the ReVerb open information extraction (IE) system (Fader et al., 2011) on the news streams to obtain a set of (a_1, r, a_2, t) tuples, where the a_i are the arguments, r is a relation phrase, and t is the time-stamp of the corresponding news article. When (a_1, a_2, t) suggests a real word event, the relation r of (a_1, r, a_2, t) is likely to describe that event (*e.g. (Armstrong, resign from, Livestrong, Oct 17*). We call every (a_1, a_2, t) an extracted event candidate (EEC), and every relation describing the event an *event-mention*.

For each EEC (a_1, a_2, t), suppose there are m extraction tuples $(a_1, r_1, a_2, t) \ldots (a_1, r_m, a_2, t)$ sharing the values of a_1, a_2, and t. We refer to this set of extraction tuples as the *EEC-set*, and denote it $(a_1, a_2, t, \{r_1 \ldots r_m\})$. All the event-mentions in the EEC-set may be semantically equivalent and are hence candidates for a good paraphrase cluster.

Thus, the paraphrasing problem becomes a prediction problem: for each relation r_i in the EEC-set, does it or does it not describe the hypothesized event? We solve this problem in two steps. The

[2] https://www.cs.washington.edu/node/9473/

next section proposes a set of temporal correspondence heuristics that partially characterize semantically equivalent EEC-sets. Then, in Section 4, we present a joint inference model designed to use these heuristics to solve the prediction problem and to generate paraphrase clusters.

3 Temporal Correspondence Heuristics

In this section, we propose a set of temporal heuristics that are useful to generate paraphrases at high precision. Our heuristics start from the basic observation mentioned previously — events can often be uniquely determined by their arguments and time. Additionally, we find that it is not just the *publication time* of the news story that matters, the *verb tenses* of the sentences are also important. For example, the two sentences *"Armstrong was the chairman of Livestrong"* and *"Armstrong steps down from Livestrong"* have past and present tense respectively, which suggests that the relation phrases are less likely to describe the same event and are thus not semantically equivalent. To capture these intuitions, we propose the *Temporal Functionality Heuristic*:

Temporal Functionality Heuristic. *News articles published at the same time that mention the same entities and use the same tense tend to describe the same events.*

Unfortunately, we find that not all the event candidates, (a_1, a_2, t), are equally good for paraphrasing. For example, today's news might include both *"Barack Obama heads to the White House"* and *"Barack Obama greets reporters at the White House"*. Although the two sentences are highly similar, sharing a_1 = "Barack Obama" and a_2 = "White House," and were published at the same time, they describe different events.

From a probabilistic point of view, we can treat each sentence as being generated by a particular hidden event which involves several actors. Clearly, some of these actors, like Obama, participate in many more events than others, and in such cases we observe sentences generated from a *mixture* of events. Since two event mentions from such a mixture are much less likely to denote the same event or relation, we wish to distinguish them from the better (semantically homogeneous) EECs like the *(Armstrong, Livestrong)* example. The question becomes "How one can distinguish good entity pairs from bad?"

Our method rests on the simple observation that an entity which participates in many different events on one day is likely to have participated in events in recent days. Therefore we can judge whether an entity pair is good for paraphrasing by looking at the *history of the frequencies* that the entity pair is mentioned in the news streams, which is the *time series* of that entity pair. The time series of the entity pair *(Barack Obama, the White House)* tends to be high over time, while the time series of the entity pair *(Armstrong, Livestrong)* is flat for a long time and suddenly spikes upwards on a single day. This observation leads to:

Temporal Burstiness Heuristic. *If an entity or an entity pair appears significantly more frequently in one day's news than in recent history, the corresponding event candidates are likely to be good to generate paraphrase.*

The temporal burstiness heuristic implies that a good EEC (a_1, a_2, t) tends to have a *spike* in the time series of its entities a_i, or argument pair (a_1, a_2), on day t.

However, even if we have selected a good EEC for paraphrasing, it is likely that it contains a few relation phrases that are related to (but not synonymous with) the other relations included in the EEC. For example, it's likely that the news story reporting *"Armstrong steps down from Livestrong."* might also mention *"Armstrong is the founder of Livestrong."* and so both "steps down from" and "is the founder of" relation phrases would be part of the same EEC-set. Inspired by the idea of one sense per discourse from (Gale et al., 1992), we propose:

One Event-Mention Per Discourse Heuristic. *A news article tends not to state the same fact more than once.*

The one event-mention per discourse heuristic is proposed in order to gain precision at the expense of recall — the heuristic directs an algorithm to choose, from a news story, the single "best" relation phrase connecting a pair of two entities. Of course, this doesn't answer the question of deciding which phrase is "best." In Section 4.3, we describe how to learn a probabilistic graphical model which does exactly this.

4 Exploiting the Temporal Heuristics

In this section we propose several models to capture the temporal correspondence heuristics, and discuss their pros and cons.

4.1 Baseline Model

An easy way to use an EEC-set is to simply predict that all r_i in the EEC-set are event-mentions, and hence are semantically equivalent. That is, given EEC-set $(a_1, a_2, t, \{r_1 \dots r_m\})$, the output cluster is $\{r_1 \dots r_m\}$.

This baseline model captures the most of the temporal functionality heuristic, except for the tense requirement. Our empirical study shows that it performs surprisingly well. This demonstrates that the quality of our input for the learning model is good: the EEC-sets are promising resources for paraphrasing.

Unfortunately, the baseline model cannot deal with the other heuristics, a problem we will remedy in the following sections.

4.2 Pairwise Model

The temporal functionality heuristic suggests we exploit the tenses of the relations in an EEC-set; while the temporal burstiness heuristic suggests we exploit the time series of its arguments. A pairwise model can be designed to capture them: we compare pairs of relations in the EEC-set, and predict whether each pair is synonymous or non-synonymous. Paraphrase clusters are then generated according to some heuristic rules (*e.g.* assuming transitivity among synonyms). The tenses of the relations and time series of the arguments are encoded as features, which we call *tense features* and *spike features* respectively. An example tense feature is whether one relation is past tense while the other relation is present tense; an example spike feature is the covariance of the time series.

The pairwise model can be considered similar to paraphrasing techniques which examine two sentences and determine whether they are semantically equivalent (Dolan and Brockett, 2005; Socher et al., 2011). Unfortunately, these techniques often based purely on text metrics and does not consider any temporal attributes. In section 5, we evaluate the effect of applying these techniques.

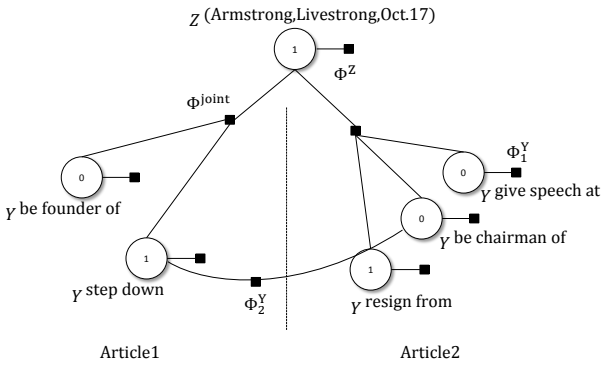

Figure 2: an example model for EEC (Armstrong, Livestrong, Oct 17). Y and Z are binary random variables. Φ^Y, Φ^Z and Φ^{joint} are factors. *be founder of* and *step down* come from article 1 while *give speech at*, *be chairman of* and *resign from* come from article 2.

4.3 Joint Cluster Model

The pairwise model has several drawbacks: *1)* it lacks the ability to handle constraints, such as the mutual exclusion constraint implied by the one-mention per discourse heuristic; *2)* ad-hoc rules, rather than formal optimizations, are required to generate clusters containing more than two relations.

A common approach to overcome the drawbacks of the pairwise model and to combine heuristics together is to introduce a joint cluster model, in which heuristics are encoded as features and constraints. Data, instead of ad-hoc rules, determines the relevance of different insights, which can be learned as parameters. The advantage of the joint model is analogous to that of cluster-based approaches for coreference resolution (CR). In particular, a joint model can better capture constraints on multiple variables and can yield higher quality results than pairwise CR models (Rahman and Ng, 2009).

We propose an undirected graphical model, NEWSSPIKE, which jointly clusters relations. Constraints are captured by factors connecting multiple random variables. We introduce random variables, the factors, the objective function, the inference algorithm, and the learning algorithm in the following sections. Figure 2 shows an example model for EEC *(Armstrong, Livestrong, Oct 17)*.

4.3.1 Random Variables

For the EEC-set $(a_1, a_2, t, \{r_1, \dots r_m\})$, we introduce one event variable and m relation variables, all boolean valued. The event variable $Z^{(a_1,a_2,t)}$ indi-

cates whether (a_1, a_2, t) is a good event for paraphrasing. It is designed in accordance with the temporal burstiness heuristic: for the EEC *(Barack Obama, the White House, Oct 17)*, Z should be assigned the value 0.

The relation variable Y^r indicates whether relation r describes the EEC (a_1, a_2, t) or not (*i.e.* r is an event-mention or not). The set of all event-mentions with $Y^r = 1$ define a paraphrase cluster, containing relation phrases. For example, the assignments $Y^{\textit{step down}} = Y^{\textit{resign from}} = 1$ produce a paraphrase cluster {*step down, resign from*}.

4.3.2 Factors and the Joint Distribution

In this section, we introduce a conditional probability model defining a joint distribution over all of the event and relation variables. The joint distribution is a function over *factors*. Our model contains *event factors*, *relation factors* and *joint factors*.

The event factor Φ^Z is a log-linear function with spike features, used to distinguish good events. A relation factor Φ^Y is also a log-linear function. It can be defined for individual relation variables (*e.g.* Φ_1^Y in Figure 2) with features such as whether a relation phrase comes from a clausal complement[3]. A relation factor can also be defined for a pair of relation variables (*e.g.* Φ_2^Y in Figure 2) with features capturing the pairwise evidence for paraphrasing, such as if two relation phrases have the same tense.

The joint factors Φ^{joint} are defined to apply constraints implied by the temporal heuristics. They play two roles in our model: *1)* to satisfy the temporal burstiness heuristic, when the value of the event variable is false, the EEC is not appropriate for paraphrasing, and so all relation variables should also be false; and *2)* to satisfy the one-mention per discourse heuristic, at most one relation variable from a single article could be true.

We define the joint distribution over these variables and factors as follows. Let $\mathbf{Y} = (Y^{r_1} \ldots Y^{r_m})$ be the vector of relation variables; let $\mathbf{x}$ be the features. The joint distribution is:

$$p(Z = z, \mathbf{Y} = \mathbf{y}|\mathbf{x}; \Theta) \stackrel{\text{def}}{=} \frac{1}{Z_x} \Phi^Z(z, \mathbf{x}) \times \prod_d \Phi^{\text{joint}}(z, \mathbf{y}_d, \mathbf{x}) \prod_{i,j} \Phi^Y(y_i, y_j, \mathbf{x})$$

where $\mathbf{y}_d$ indicates the subset of relation variables from a particular article d, and the parameter vector Θ is the weight vector of the features in Φ^Z and Φ^Y, which are log-linear functions; *i.e.*,

$$\Phi^Y(y_i, y_j, \mathbf{x}) \stackrel{\text{def}}{=} \exp\left(\sum_j \theta_j \phi_j(y_i, y_j, \mathbf{x})\right)$$

where ϕ_j is the jth feature function.

The joint factors Φ^{joint} are used to apply the temporal burstiness heuristic and the one event-mention per discourse heuristic. Φ^{joint} is zero when the EEC is not good for paraphrasing, but some $y^r = 1$; or when there is more than one r in a single article such that $y^r = 1$. Formally, it is calculated as:

$$\Phi^{\text{joint}}(z, \mathbf{y}_d, \mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } z = 0 \wedge \exists y^r = 1 \\ 0 & \text{if } \sum_{y^r \in \mathbf{y}_d} y^r > 1 \\ 1 & \text{otherwise} \end{cases}$$

4.3.3 Maximum a Posteriori Inference

The goal of inference is to find the predictions $z, \mathbf{y}$ which yield the greatest probability, *i.e.*,

$$z^*, \mathbf{y}^* = \arg\max_{z,\mathbf{y}} p(Z = z, \mathbf{Y} = \mathbf{y}|\mathbf{x}; \Theta)$$

This can be viewed as a MAP inference problem. In general, inference in a graphical model is challenging. Fortunately, the joint factors in our model are linear, and the event and relation factors are log-linear; we can cast MAP inference as an integer linear programming (ILP) problem, and then compute an approximation in polynomial time by means of linear programming using randomized rounding, as proposed in (Yannakakis, 1992).

We build one ILP problem for every EEC. The variables of the ILP are Z and $\mathbf{Y}$, which only take values of 0 or 1. The objective function is the sum of logs of the event and relation factors Φ^Z and Φ^Y. The temporal burstiness heuristic of Φ^{joint} is encoded as a linear inequality constraint $z \geq y_i$; the one-mention per discourse heuristic of Φ^{joint} is encoded as the constraint $\sum_{y_i \in \mathbf{y}_d} y_i \leq 1$.

[3]Relation phrases in clausal complement are less useful for paraphrasing because they often do not describe a fact. For example, in the sentence *He heard Romney had won the election*, the extraction (Romney, had won, the election) is not a fact at all.

4.3.4 Learning

Our training data consists a set of $N = 500$ labeled EEC-sets each in the form of $\{(R_i, R_i^{\text{gold}}) \mid_{i=1}^{N}\}$. Each R is the set of all relations in the EEC-set while R^{gold} is a manually selected subset of R containing relations describing the EEC. R^{gold} could be empty if the EEC was deemed poor for paraphrasing. For our model, the gold assignment $y^{r\,\text{gold}} = 1$ if $r \in R^{\text{gold}}$; the gold assignment $z^{\text{gold}} = 1$ if R^{gold} is not empty.

Given $\{(R_i, R_i^{\text{gold}}) \mid_{i=1}^{N}\}$, learning over similar models is commonly done via maximum likelihood estimation as follows:

$$L(\Theta) = \log \prod_i p(Z_i = z_i^{\text{gold}}, \mathbf{Y}_i = \mathbf{y}_i^{\text{gold}} \mid \mathbf{x}_i, \Theta)$$

For features in relation factors, the partial derivative for the ith model is:

$$\Phi_j(\mathbf{y}_i^{\text{gold}}, \mathbf{x}_i) - E_{p(z_i, \mathbf{y}_i|, \mathbf{x}_i, \Theta)} \Phi_j(\mathbf{y}_i, \mathbf{x}_i)$$

where $\Phi_j(\mathbf{y}_i, \mathbf{x}_i) = \sum \phi_j(X, Y, \mathbf{x})$, the sum of values for the jth feature in the ith model; and values of X, Y come from the assignment $\mathbf{y}_i$. For features in event factors, the partial derivative is derived similarly as

$$\phi_j(z_i^{\text{gold}}, \mathbf{x}_i) - E_{p(z_i, \mathbf{y}_i|, \mathbf{x}_i, \Theta)} \phi_j(z_i, \mathbf{x}_i)$$

It is unclear how to efficiently compute the expectations in the above formula, a brute force approach requires enumerating all assignments of $\mathbf{y}_i$, which is exponentially large with the number of relations. Instead, we opt to use a more tractable perceptron learning approach (Collins, 2002; Hoffmann et al., 2011). Instead of computing the expectations, we simply compute $\phi_j(z_i^*, \mathbf{x}_i)$ and $\Phi_j(\mathbf{y}_i^*, \mathbf{x}_i)$, where $z_i^*, \mathbf{y}_i^*$ is the assignment with the highest probability, generated by the MAP inference algorithm using the current weight vector. The weight updates are the following:

$$\Phi_j(\mathbf{y}_i^{\text{gold}}, \mathbf{x}_i) - \Phi_j(\mathbf{y}_i^*, \mathbf{x}_i) \quad (1)$$

$$\phi_j(z_i^{\text{gold}}, \mathbf{x}_i) - \phi_j(z_i^*, \mathbf{x}_i) \quad (2)$$

The updates can be intuitively explained as penalties on errors. In sum, our learning algorithm consists of iterating the following two steps: (1) infer the most probable assignment given the current weights; (2) update the weights by comparing inferred assignments and the truth assignment.

5 Empirical Study

We first introduce the experimental setup for our empirical study, and then we attempt to answer two questions in sections 5.2 and 5.3 respectively: First, does the NEWSSPIKE algorithm effectively exploit the proposed heuristics and outperform other approaches which also use news streams? Secondly, do the proposed temporal heuristics paraphrase relations with greater precision than the distributional hypothesis?

5.1 Experimental Setup

Since we were unable to find any elaborate time-stamped, parallel, news corpus, we collected data using the following procedure:

- Collect RSS news seeds, which contain the title, time-stamp, and abstract of the news items.
- Use these titles to query the Bing news search engine API and collect additional time-stamped news articles.
- Strip HTML tags from the news articles using Boilerpipe (Kohlschütter et al., 2010); keep only the title and first paragraph of each article.
- Extract shallow relation tuples using the OpenIE system (Fader et al., 2011).

We performed these steps every day from January 1 to February 22, 2013. In total, we collected 546,713 news articles, for which 2.6 million extractions had 529 thousand unique relations.

We used several types of features for paraphrasing: *1)* spike features obtained from time series; *2)* tense features, such as whether two relation phrases are both in the present tense; *3)* cause-effect features, such as whether two relation phrases often appear successively in the news articles; *4)* text features, such as whether sentences are similar; *5)* syntactic features, such as whether a relation phrase appears in a clausal complement; and *6)* semantic features, such as whether a relation phrase contains negative words.

Text and semantic features are encoded using the relation factors of section 4.3.2. For example, in Figure 2, the factor Φ_2^Y includes the textual similarity between the sentences containing the phrases *"step down"* and *"be chairman of"* respectively; it also includes the feature that the tense of *"step down"* (present) is different from the tense of *"be chairman*

output	{**go into**, **go to**, *speak*, *return*, **head to**}
gold	{**go into**, **go to**, *approach*, **head to**}
gold$_{\text{div}}$	{**go** $*$, *approach*, **head to**}
P/R	precision $= 3/5$ recall $= 3/4$
P/R$_{\text{div}}$	precision$_{\text{div}} = 2/4$ recall$_{\text{div}} = 2/3$

Figure 3: an example pair of the output cluster and the gold cluster, and the corresponding precision recall numbers.

of" (past).

5.2 Comparison with Methods using Parallel News Corpora

We evaluated NEWSSPIKE against other methods that also use time-stamped news. These include the models mentioned in section 3 and state-of-the-art paraphrasing techniques.

Human annotators created gold paraphrase clusters for 500 EEC-sets; note that some EEC-sets yield no gold cluster, since at least two synonymous phrases. Two annotators were shown a set of candidate relation phrases in context and asked to select a subset of these that described a shared event (if one existed). There was 98% phrase-level agreement. Precision and recall were computed by comparing an algorithm's output clusters to the gold cluster of each EEC. We consider paraphrases with minor lexical diversity, *e.g. (go to, go into)*, to be of lesser interest. Since counting these trivial paraphrases tends to exaggerate the performance of a system, we also report precision and recall on *diverse clusters i.e.*, those whose relation phrases all have different head verbs. Figure 3 illustrates these metrics with an example; note under our diverse metrics, all phrases matching *go* * count as one when computing both precision and recall. We conduct 5-fold cross validation on our labeled dataset to get precision and recall numbers when the system requires training.

We compare NEWSSPIKE with the models in Section 4, and also with the state-of-the-art paraphrase extraction method:

Baseline: the model discussed in Section 4.1. This system does not need any training, and generates outputs with perfect recall.

Pairwise: the pairwise model discussed in Section 4.2 and using the same set of features as used by NEWSSPIKE. To generate output clusters, transitivity is assumed inside the EEC-set. For example, when the pairwise model predicts that (r_1, r_2) and (r_1, r_3) are both paraphrases, the resulting cluster is $\{r_1, r_2, r_3\}$.

System	P/R		P/R diverse	
	prec	rec	prec	rec
Baseline	0.67	1.00	0.53	1.00
Pairwise	0.90	0.60	0.81	0.37
Socher	0.81	0.35	0.68	0.29
NEWSSPIKE	**0.92**	0.55	**0.87**	0.31

Table 1: Comparison with methods using parallel news corpora

Socher: Socher *et al.* (2011) achieved the best results on the Dolan *et al.* (2004) dataset, and released their code and models. We used their off-the-shelf predictor to replace the classifier in our Pairwise model. Given sentential paraphrases, aligning relation phrases is natural, because OpenIE has already identified the relation phrases.

Table 1 shows precision and recall numbers. It is interesting that the basic model already obtains 0.67 precision overall and 0.53 in the diverse condition. This demonstrates that the EEC-sets generated from the news streams are a promising resource for paraphrasing. Socher's method performs better, but not as well as Pairwise or NEWSSPIKE, especially in the diverse cases. This is probably due to the fact that Socher's method is based purely on text metrics and does not consider any temporal attributes. Taking into account the features used by NEWSSPIKE, Pairwise significantly improves the precision, which demonstrates the power of our temporal correspondence heuristics. Our joint cluster model, NEWSSPIKE, which considers both temporal features and constraints, gets the best performance in both conditions.

We conducted ablation testing to evaluate how spike features and tense features, which are particularly relevant to the temporal aspects of news streams, can improve performance. Figure 4 compares the precision/recall curves for three systems in the diverse condition: (1) NEWSSPIKE; (2) w/oSpike: turning off all spike features; and (3) w/oTense: turning off all features about tense. (4) w/oDiscourse: turning off one event-mention per discourse heuristic. There are some dips in

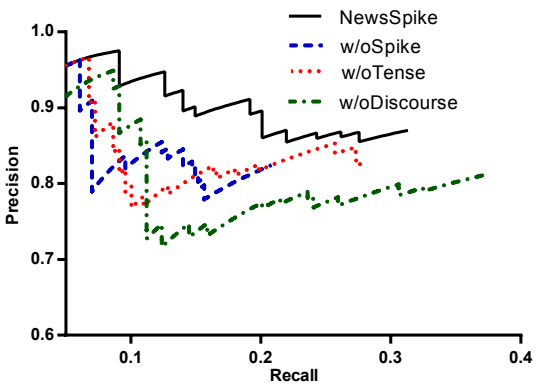

Figure 4: Precision recall curves on hard, diverse cases for NewsSpike, w/oSpike, w/oTense and w/oDiscourse.

the curves because they are drawn after sorting the predictions by the value of the corresponding ILP objective functions, which do not perfectly reflect prediction accuracy. However, it is clear that NEWSSPIKE produces greater precision over all ranges of recall.

5.3 Comparison with Methods using the Distributional Hypothesis

We evaluated our model against methods based on the distributional hypothesis. We ran NEWSSPIKE over all EEC-sets except for the development set and compared to the following systems:

Resolver: Resolver (Yates and Etzioni, 2009) uses a set of extraction tuples in the form of (a_1, r, a_2) as the input and creates a set of relation clusters as the output paraphrases. Resolver also produces argument clusters, but this paper only evaluates relation clustering. We evaluated Resolver's performance with an input of the 2.6 million extractions described in section 5.1, using Resolver's default parameters.

ResolverNYT: Since Resolver is supposed to perform better when given more accurate statistics from a larger corpus, we tried giving it more data. Specifically, we ran ReVerb on 1.8 million NY Times articles published between 1987 and 2007 obtain 60 million extractions (Sandhaus, 2008). We ran Resolver on the union of this and our standard test set, but report performance only on clusters whose relations were seen in our news stream.

System	all		diverse	
	prec	#rels	prec	#rels
Resolver	0.78	129	0.65	57
ResolverNyt	0.64	1461	0.52	841
ResolverNytTop	0.83	207	0.72	79
Cosine	0.65	17	0.33	9
CosineNyt	0.56	73	0.46	59
NEWSSPIKE	**0.93**	24843	**0.87**	5574

Table 2: Comparison with methods using the distributional hypothesis

ResolverNytTop: Resolver is designed to achieve good performance on its top results. We thus ranked the ResolverNYT outputs by their scores and report the precision of the top 100 clusters.

Cosine: Cosine similarity is a basic metric for the distributional hypothesis. This system employs the same setup as Resolver in order to generate paraphrase clusters, except that Resolver's similarity metric is replaced with the cosine. Each relation is represented by a vector of argument pairs. The similarity threshold to merge two clusters was 0.5.

CosineNYT: As for ResolverNYT, we ran CosineNYT with an extra 60 million extractions and reported the performance on relations seen in our news stream.

We measured the precision of each system by manually labeling all output if 100 or fewer clusters were generated (*e.g.* ResolverNytTop), otherwise 100 randomly chosen clusters were sampled. Annotators first determined the meaning of every output cluster and then created a gold cluster by choosing the correct relations. The gold cluster could be empty if the output cluster was nonsensical. Unlike many papers that simply report recall on the most frequent relations, we evaluated the total number of returned relations in the output clusters. As in Section 5.2, we also report numbers for the case of lexically diverse relation phrases.

As can be seen in Table 2, NEWSSPIKE outperformed methods based on the distributional hypothesis. The performance of the Cosine and CosineNyt was very low, suggesting that simple similarity metrics are insufficient for handling the paraphrasing problem, even when large-scale input is involved. Resolver and ResolverNyt employ an advanced similarity measurement and achieve better results. However, it is surprising that Resolver results in a greater precision than ResolverNyt. It

is possible that argument pairs from news streams spanning 20 years sometimes provide incorrect evidence for paraphrasing. For example, there were extractions like *(the Rangers, be third in, the NHL)* and *(the Rangers, be fourth in, the NHL)* from news in 2007 and 2003 respectively. Using these phrases, ResolverNyt produced the incorrect cluster {*be third in, be fourth in*}. NEWSSPIKE achieves greater precision than even the best results from ResolverNyt-Top, because NEWSSPIKE successfully captures the temporal heuristics, and does not confuse synonyms with antonyms, or causes with effects. NEWSSPIKE also returned on order of magnitude more relations than other methods.

5.4 Discussion

Unlike some domain-specific clustering methods, we tested on all relation phrases extracted by OpenIE on the collected news streams. There are no restrictions on the types of relations. Output paraphrases cover a broad range, including politics, sports, entertainment, health, science, etc. There are 10 thousand nonempty clusters over 17 thousand distinct phrases with average size 2.4. Unlike methods based on distributional similarity, NewsSpike correctly clusters infrequently appearing phrases.

Since we focus on high precision, it is not surprising that most clusters are of size 2 and 3. These high precision clusters can contribute a lot to generate larger paraphrase clusters. For example, one can invent the technique to merge smaller clusters together. The work presented here provides a foundation for future work to more closely examine these challenges.

While this paper gives promising results, there are still behaviors found in news streams that prove challenging. Many errors are due to the discourse context: the two sentences are synonymous in the given EEC-set, but the relation phrases are not paraphrases in general. For example, consider the following two sentences: *"DA14 narrowly misses Earth"* and *"DA14 flies so close to Earth"*. Statistics information from large corpus would be helpful to handle such challenges. Note in this paper, in order to fairly compare with the distributional hypothesis, we purposely forced NEWSSPIKE not to rely on any distributional similarity. But NEWSSPIKE's graphical model has the flexibility to incorporate any similarity metrics as features. Such a hybrid model has great potential to increase both precision and recall, which is one goal for future work.

6 Related Work

The vast majority of paraphrasing work falls into two categories: approaches based on the distributional hypothesis or those exploiting on correspondences between parallel corpora (Androutsopoulos and Malakasiotis, 2010; Madnani and Dorr, 2010).

Using Distribution Similarity: Lin and Pantel's (2001) DIRT employ mutual information statistics to compute the similarity between relations represented in dependency paths. Resolver (Yates and Etzioni, 2009) introduces a new similarity metric called the Extracted Shared Property (ESP) and uses a probabilistic model to merge ESP with surface string similarity.

Identifying the semantic equivalence of relation phrases is also called *relation discovery* or *unsupervised semantic parsing*. Often techniques don't compute the similarity explicitly but rely implicitly on the distributional hypothesis. Poon and Domingos' (2009) USP clusters relations represented with fragments of dependency trees by repeatedly merging relations having similar context. Yao *et al.* (2011; 2012) introduces generative models for relation discovery using LDA-style algorithm over a relation-feature matrix. Chen *et al.* (2011) focuses on domain-dependent relation discovery, extending a generative model with meta-constraints from lexical, syntactic and discourse regularities.

Our work solves a major problem with these approaches, avoiding errors such as confusing synonyms with antonyms and causes with effects. Furthermore, NEWSSPIKE doesn't require massive statistical evidence as do most approaches based on the distributional hypothesis.

Using Parallel Corpora: Comparable and parallel corpora, including news streams and multiple translations of the same story, have been used to generate paraphrases, both sentential (Barzilay and Lee, 2003; Dolan et al., 2004; Shinyama and Sekine, 2003) and phrasal (Barzilay and McKeown, 2001; Shen et al., 2006; Pang et al., 2003). Typical methods first gather relevant articles and then pair sentences that are potential paraphrases. Given a training set of paraphrases, models are learned and applied to unlabeled pairs (Dolan and Brockett, 2005;

Socher et al., 2011). Phrasal paraphrases are often obtained by running an alignment algorithm over the paraphrased sentence pairs.

While prior work uses the temporal aspects of news streams as a coarse filter, it largely relies on text metrics, such as context similarity and edit distance, to make predictions and alignments. These metrics are usually insufficient to produce high precision results; moreover they tend to produce paraphrases that are simple lexical variants (*e.g.* {*go to, go into*}.). In contrast, NEWSSPIKE generates paraphrase clusters with both high precision and high diversity.

Others: Textual entailment (Dagan et al., 2009), which finds a phrase implying another phrase, is closely related to the paraphrasing task. Berant *et al.* (2011) notes the flaws in distributional similarity and proposes local entailment classifiers, which are able to combine many features. Lin *et al.* (2012) also uses temporal information to detect the semantics of entities. In a manner similar to our approach, Recasens *et al.* (2013) mines parallel news stories to find opaque coreferent mentions.

7 Conclusion

Paraphrasing event relations is crucial to many natural language processing applications, including relation extraction, question answering, summarization, and machine translation. Unfortunately, previous approaches based on distribution similarity and parallel corpora, often produce low precision clusters. This paper introduces three Temporal Correspondence Heuristics that characterize semantically equivalent phrases in news streams. We present a novel algorithm, NEWSSPIKE, based on a probabilistic graphical model encoding these heuristics, which harvests high-quality paraphrases of event relations.

Experiments show NEWSSPIKE's improvement relative to several other methods, especially at producing lexically diverse clusters. Ablation tests confirm that our temporal features are crucial to NEWSSPIKE's precision. In order to spur future research, we are releasing an annotated corpus of time-stamped news articles and our harvested relation clusters.

Acknowledgments

We thank Oren Etzioni, Anthony Fader, Raphael Hoffmann, Ben Taskar, Luke Zettlemoyer, and the anonymous reviewers for providing valuable advice. We also thank Shengliang Xu for annotating the datasets. We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181, ONR grant N00014-12-1-0211, a gift from Google, and the WRF / TJ Cable Professorship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. In *Journal of Artificial Intelligence Research*, pages 135–187.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL*, pages 16–23. Association for Computational Linguistics.

Regina Barzilay and Kathleen R McKeown. 2001. Extracting paraphrases from a parallel corpus. In *ACL*, pages 50–57. Association for Computational Linguistics.

Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *ACL*, pages 550–557. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *ACL-HLT*, pages 610–619. Association for Computational Linguistics.

Rahul Bhagat and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In *ACL*, volume 8, pages 674–682. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *NAACL*, pages 17–24. Association for Computational Linguistics.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. 2011. In-domain relation discovery with meta-constraints via posterior regularization. In *ACL-HLT*, pages 530–540. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *ACL*, pages 1–8. Association for Computational Linguistics.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(04):i–xvii.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of IWP*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Computational Linguistics*, page 350. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*. Association for Computational Linguistics, July 27-31.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *ACL*. Association for Computational Linguistics.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.

Zellig S Harris. 1954. Distributional structure. *Word*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL-HLT*, pages 541–550.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *WSDM*, pages 441–450. ACM.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Thomas Lin, Oren Etzioni, et al. 2012. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP*, pages 893–903. Association for Computational Linguistics.

Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *NAACL*, pages 102–109. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *EMNLP*, pages 1–10. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *EMNLP*, pages 968–977. Association for Computational Linguistics.

Marta Recasens, Matthew Can, and Dan Jurafsky. 2013. Same referent, different words: Unsupervised mining of opaque coreferent mentions. In *Proceedings of NAACL-HLT*, pages 897–906.

Evan Sandhaus. 2008. *The New York Times annotated corpus*. Linguistic Data Consortium.

Siwei Shen, Dragomir R Radev, Agam Patel, and Güneş Erkan. 2006. Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 747–754. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2003. Paraphrase acquisition for information extraction. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 65–71. Association for Computational Linguistics.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *NIPS*, 24:801–809.

Mihalis Yannakakis. 1992. On the approximation of maximum satisfiability. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, SODA '92, pages 1–9.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *EMNLP*, pages 1456–1466. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *ACL*, pages 712–720. Association for Computational Linguistics.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255.