

Paraphrasing vs Coreferring: Two Sides of the Same Coin

Yehudit Meged¹ Avi Caciularu^{1,2} Vered Shwartz^{2,3} Ido Dagan¹

¹Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel

²Allen Institute for Artificial Intelligence

³Paul G. Allen School of Computer Science & Engineering, University of Washington

{yehuditmeged, avi.c33}@gmail.com, vereds@allenai.org, dagan@cs.biu.ac.il

Abstract

We study the potential synergy between two different NLP tasks, both confronting predicate lexical variability: identifying predicate paraphrases, and event coreference resolution. First, we used annotations from an event coreference dataset as distant supervision to re-score heuristically-extracted predicate paraphrases. The new scoring gained more than 18 points in average precision upon their ranking by the original scoring method. Then, we used the same re-ranking features as additional inputs to a state-of-the-art event coreference resolution model, which yielded modest but consistent improvements to the model’s performance. The results suggest a promising direction to leverage data and models for each of the tasks to the benefit of the other.

1 Introduction

Recognizing that mentions of different lexical predicates discuss the same event is challenging (Barhom et al., 2019). Lexical resources such as WordNet (Miller, 1995) capture such synonyms (*say, tell*) and hypernyms (*whisper, talk*), as well as antonyms, which can be used to refer to the same event when the arguments are reversed ([a]₀ beat [a]₁, [a]₁ lose to [a]₀). However, WordNet’s coverage is insufficient, in particular, missing context-specific paraphrases (e.g. (*hide, launder*), in the context of money). Conversely, distributional methods enjoy broader coverage, but their precision for this purpose is limited because distributionally similar terms may often be mutually-exclusive (*born, die*) or may refer to different event types which are only temporally or causally related (*sentenced, convicted*).

Two prominent lines of work pertaining to identifying predicates whose meaning or referents can be matched are cross-document (CD) event coreference resolution and recognizing predicate para-

Tara Reid has checked into_✓ Promises Treatment Center.
Actress Tara Reid entered_✓ well-known Malibu rehab center.
Lindsay Lohan checked into_× rehab in Malibu, California.

Director Chris Weitz is expected to direct_✓ New Moon.
Chris Weitz will take on_✓ the sequel to “Twilight”.
Gary Ross is still in negotiations to direct_× the sequel.

Table 1: Examples from ECB+ (a cross-document coreference dataset) that illustrate the context-sensitive nature of event coreference. The illustrated predicates are co-referable, and hence may be used to refer to the same event in certain contexts, but obviously not all their mentions corefer.

phrases. The former identifies and clusters event mentions, across multiple documents, that refer to the same event *within their respective contexts*. The latter task, on the other hand, collects pairs of event expressions that, at the generic lexical level, may refer to the same event *in certain contexts*. Table 1 illustrates this difference with examples of co-referable predicate paraphrases, while their mentions obviously do not always co-refer.

Cross-document event coreference resolution systems are typically supervised, usually trained on the ECB+ dataset, which contains clusters of news articles on different topics (Cybulska and Vossen, 2014). Recent systems rely on neural representations of the mentions and their contexts (Kenyon-Dean et al., 2018; Barhom et al., 2019), while earlier approaches leveraged WordNet and other lexical resources to obtain a signal of whether a pair of mentions *may* be coreferring (e.g. Bejan and Harabagiu, 2010; Yang et al., 2015).

Approaches for acquiring predicate paraphrase, in the form of a pair of paraphrastic predicates or predicate templates, were based mostly on unsupervised signals. These included similarity between argument distributions (Lin and Pantel, 2001; Berant, 2012), backtranslation across languages (Barzilay and McKeown, 2001; Ganitkevitch et al., 2013;

Mallinson et al., 2017), or leveraging redundant news reports on the same event, which are hence likely to refer to the same events and entities using different words (Shinyama et al., 2002; Shinyama and Sekine, 2006; Barzilay and Lee, 2003; Zhang and Weld, 2013; Xu et al., 2014; Shwartz et al., 2017). In some cases, the paraphrase collection phase includes a step of validating a subset of the paraphrases and training a model on these gold paraphrases to re-rank the entire resource (Lan et al., 2017).

In this paper, we study the potential synergy between predicate paraphrases and event coreference resolution. We show that the data and models for one task can benefit the other. In one direction (Section 3), we use event coreference annotations from the ECB+ dataset as distant supervision to learn an improved scoring of predicate paraphrases in the unsupervised Chirps resource (Shwartz et al., 2017). The distantly supervised scorer significantly improves upon ranking by the original Chirps scores, adding 18 points to average precision over a test sample.

In the other direction (Section 4), we incorporate data from Chirps, represented in the Chirps re-scorer feature vector, into a state-of-the-art event coreference system (Barhom et al., 2019). Chirps has a substantial coverage over the ECB+ coreferencing mention pairs, and consequently, the incorporation yields a modest but consistent improvement across the various coreference metrics.¹

2 Background and Motivation

In this section we provide some background about the cross-document coreference resolution and paraphrase identification (acquisition) tasks, which is relevant for our approaches for synergizing these two tasks.

2.1 Event Coreference Resolution

Event coreference resolution aims to identify and cluster event mentions, that, within their respective contexts, refer to the same event. The task has two variants, one in which coreferencing mentions are within the same document (within document) and another in which coreferencing mentions may be in different documents (cross-document, CD), on which we focus in this paper.

¹Code available at github.com/yehudit96/coreferrability, github.com/yehudit96/event_entity_coref_ecb_plus

The standard datasets used for CD event coreference training and evaluation are ECB+ (Cybulska and Vossen, 2014), and its predecessors, EECB (Lee et al., 2012) and ECB (Bejan and Harabagiu, 2010). ECB+ contains a set of topics, each containing a set of documents describing the same global event. Both event and entity coreferences are annotated in ECB+, within and across documents.

Models for CD event coreference utilize a range of features, including lexical overlap among mention pairs and semantic knowledge from WordNet (Bejan and Harabagiu, 2010, 2014; Yang et al., 2015), distributional (Choubey and Huang, 2017) and contextual representations (Kenyon-Dean et al., 2018; Barhom et al., 2019).

The current state-of-the-art model from Barhom et al. (2019) iteratively and intermittently learns to cluster events and entities. A mention representation m_i consists of several components, representing both the mention span and its surrounding context. The interdependence between clustering event vs. entity mentions is encoded into the mention representation, such that an event mention representation contains a component reflecting the current entity clustering, and vice versa. Using this representation, the model trains a pairwise mention scoring function that predicts the probability that two mentions refer to the same event.

2.2 Paraphrase Identification and acquisition

Paraphrases are differing textual realizations of the same meaning (Ganitkevitch et al., 2013), typically phrases or sentences (Dolan et al., 2005). A prominent approach for identifying and collecting paraphrases, *backtranslation*, assumes that if two (say) English phrases translate to the same term in a foreign language, across multiple foreign languages, this indicates that these two phrases are paraphrases. This approach was first suggested by Barzilay and McKeown (2001), later adapted to acquire the large PPDB resource (Ganitkevitch et al., 2013), and was also shown to work well with neural machine translation (Mallinson et al., 2017).

Paraphrase Identification through Event Coreference. An alternative approach for paraphrase identification, on which we focus in this paper, leverages multiple news documents discussing the same event. The underlying assumption is that such redundant texts may refer to the same entities or events using lexically-divergent mentions. Coreferencing mentions are identified heuristically and

extracted as candidate paraphrases. When long documents are used, the first step in this approach is to align each pair of documents by sentences. This was done by finding sentences with shared named entities (Shinyama et al., 2002) or lexical overlap (Barzilay and Lee, 2003; Shinyama and Sekine, 2006), and by aligning pairs of predicates or arguments (Zhang and Weld, 2013; Recasens et al., 2013). In more recent work, Xu et al. (2014) and Lan et al. (2017) extracted sentential paraphrases from Twitter by heuristically matching pairs of tweets discussing the same topic.

Predicate Paraphrases. In contrast to sentential paraphrases, it is also beneficial to identify differing textual *templates* of the same meaning. In this paper we focus on binary predicate paraphrases such as (“[a₀] quit from [a₁]”, “[a₀] resign from [a₁]”).

Earlier approaches for acquiring predicate paraphrases considered a pair of predicate templates as paraphrases if the distributions of their argument instantiations were similar. For instance, in “[a₀] quit from [a₁]”, [a₀] would typically be instantiated by people names while [a₁] by employer organizations or job titles. A paraphrastic template like “[a₀] resign from [a₁]” is hence expected to have similar argument distributions, and can thus be detected by a distributional similarity approach (Lin and Pantel, 2001; Szpektor et al., 2004; Berant, 2012). Yet, as mentioned earlier, predicates with similar argument distributions are not necessarily paraphrastic, which introduces a substantial level of noise when acquiring paraphrase pairs using this approach.

In this paper, we follow the potentially more reliable paraphrase acquisition approach, which tries to heuristically identify concrete co-referring predicate mentions. Identifying such mention pairs, detected as actually being used to refer to the same event, can provide a strong signal for identifying these predicates as paraphrastic (vs. the quite noisy corpus-level signal of distributional similarity). In particular, we utilize the Chirps paraphrase acquisition method and resource, which follows this approach as described next in some detail.

Chirps: a Coreference-Driven Paraphrase Resource. Chirps (Shwartz et al., 2017) is a resource of predicate paraphrases extracted heuristically from Twitter. Chirps aims to recognize coreferring events by relying on the redundancy of

news headlines posted on Twitter on the same day. It extracts binary predicate-argument tuples from each tweet and aligns pairs of predicate mentions whose arguments match, by some lexical matching criteria. The matched pairs of arguments are termed *supporting pairs*, e.g. (*Chuck Berry*, 90) for “[Chuck Berry]₀ died at [90]₁” and “[Chuck Berry]₀ lived until [90]₁”. The predicate paraphrases, i.e. pairs of predicate templates (like “[a₀] died at [a₁]” and “[a₀] lived until [a₁]”) are then aggregated and ranked with the following (unsupervised) heuristic scoring function:

$$s = n \cdot \left(1 + \frac{d}{N}\right).$$

This score is proportional to the number of supporting pair instances in which the two templates were paired (n), as well as the number of different days in which such pairings were found (d), where N is the number of days the resource is collected. The Chirps resource provides the scored predicate paraphrases as well as the supporting pairs for each paraphrase.

Chirps has acquired more than 5 million distinct paraphrase pairs over the last 3 years. Human evaluation showed that this scoring is effective and that the percentage of correct paraphrases is higher for highly scored paraphrases. At the same time, due to the heuristic collection and scoring of predicate paraphrases in Chirps, entries in the resource may suffer from two types of errors: (1) type 1 error, i.e., the heuristic recognized pairs of non-paraphrastic predicates as paraphrases. This happens when the same arguments participate in multiple different events, as in the following paraphrases: “[Police]₀ arrest [man]₁” and “[Police]₀ shoot [man]₁”; and (2) type 2 error, when the scoring function assigned a low score to a rare but correct paraphrase pair, as in “[a₀] outgun [a₁]” and “[a₀] outperform [a₁]”, for which only a single supporting pair was found.

3 Chirps*: Leveraging Coreference Information for Paraphrasing

Our goal in this section is to improve paraphrase scoring, in the context of Chirps, while leveraging available information and methods for event cross-document coreference resolution. To that end, we introduce Chirps*, a new supervised scorer for Chirps candidate paraphrases, whose novelties are two fold. First, we extract a richer feature representation for a candidate paraphrase pair (Section 3.1), which is fed into a supervised classifier for the candidates. Second, we collect, semi-automatically,

distantly supervised training data for paraphrase classification, which is derived from the ECB+ cross-document coreference training set, leveraging the close relationship between the two tasks (Section 3.2). Finally, we provide some implementation details (Section 3.3).

3.1 Features

As described above, the original heuristic Chirps scorer relied only on a couple of features to score a candidate paraphrase pair. Our goal is to obtain a richer signal about the likelihood of a candidate predicate pair to indeed be paraphrastic. To that end, we collect a set of features from the available data, with a focus on assessing whether the instances from which the candidate pair was extracted indeed constitute cross-document coreferences.

Each candidate paraphrase pair consists of two predicate templates p_1 and p_2 , accompanied by the n supporting pair instances for the pair, each consisting a pair of argument terms, associated with this predicate paraphrase pair: $\text{support-pairs}(p_1, p_2) = \{(t_1^1, t_2^1), \dots, (t_1^n, t_2^n)\}$. Each tweet included in Chirps links to a news article, whose content we retrieve. When representing a pair of predicate templates, we include both local features (based on a single supporting pair) and global features (based on all supporting pairs).

Table 2 presents our 17 features, yielding a feature representation $f_{p_1, p_2} \in \mathcal{R}^{17}$ for a paraphrase pair, grouped by different sources of information. The first group includes features derived from the statistics provided by the original Chirps resource. The other 4 sources of information are described in the following paragraphs.

Named Entity Coverage While the original Chirps method did not utilize the content of the linked article, we find it useful to retrieve more information about the event. Specifically, it might help mitigating errors in Chirps’ argument matching mechanism, which relies on argument alignment considering only the text of the two tweets. We found that the original mechanism worked particularly well for named entities while being more error-prone for common nouns, which might require additional context.

Given $(t_1^i, t_2^i) \in \text{support-pairs}(p_1, p_2)$, we use SpaCy (Honnibal and Montani, 2017) to extract sets of named entities, NE_1 and NE_2 , from the first paragraph of the news article linked from each

tweet, respectively. We define a Named Entity Coverage score, NEC, as the maximum ratio of named entity coverage of one article by the other:

$$\text{NEC}(NE_1, NE_2) = \max\left(\frac{|NE_1 \cap NE_2|}{|NE_1|}, \frac{|NE_1 \cap NE_2|}{|NE_2|}\right)$$

We manually annotated a small balanced training set of 121 tweet pairs and used it to tune a score threshold $T = 0.26$, such that pairs of tweets whose NEC is at least T are considered coreferring. Finally, we include the following features: the number of coreferring tweet pairs (whose NEC score exceeds T) and the average NEC score of these pairs.

Cross-document Coreference Resolution We apply the state-of-the-art cross-document coreference model from Barhom et al. (2019) to data constructed such that each tweet constitutes a document and each pair of tweets corresponding to t_1^j and t_2^j in $\text{support-pairs}(p_1, p_2)$ forms a topic, to be analyzed for coreference. As input for the model, in each tweet, we mark the corresponding predicate span as an event mention and the two argument spans as entity mentions. The model outputs whether the two event mentions corefer (yielding a single event coreference cluster for the two mentions) or not (yielding two singleton clusters). Similarly, it clusters the four arguments to entity coreference clusters.

Differently from Chirps, this model makes its event clustering decision based on the predicate, arguments, and the context of the full tweet, as opposed to considering the arguments alone. Thus, we expect it not to cluster predicates whose arguments match lexically, if their contexts or predicates don’t match (first example in Table 3). In addition, the model’s mentions representation might help to identify lexically-divergent yet semantically-similar arguments (second example in Table 3).

For a given pair of tweets, we extract the following binary features with respect to the predicate mentions: *Event Perfect* when the predicates are assigned to the same cluster, and *Event No Match* when each predicate forms a singleton cluster. For argument mentions, we extract the following features: *Entity Perfect* if the two a_0 arguments belong to one cluster and the two a_1 arguments belong to another cluster; *Entity Reverse* if at least one of the a_0 arguments is clustered as coreferring with the a_1 argument in the other tweet; and *Entity No*

| | Name | Description |
|---------------------------------------|------------------------------|--|
| Chirps-based Features | # Templates | The number of different predicate paraphrase pairs with p_1 and p_2 as predicates, regardless of argument ordering, e.g. “[a ₀] release [a ₁]” / “[a ₀] reveal [a ₁]” and “release [a ₀] [a ₁]” / “reveal [a ₀] [a ₁]” are both counted. |
| | # Supporting pairs | The total number of support pairs of p_1 and p_2 across the template variants. |
| | # Days | The total number of days d that p_1 and p_2 was matched in Chirps across the template variants. |
| | # Available supporting pairs | The number of support pairs of p_1 and p_2 across the template variants <i>that were still available to download</i> . |
| | # Days of available pairs | The total number of days d in which the support pairs above occurred in the available tweets. |
| | Score | The maximal Chirps score across the template variants. |
| Named Entity Coverage | # NEC above threshold | Number of pairs with NEC score of at least T . |
| | Average above threshold | Average of NEC scores for pairs with a score of at least T . |
| Cross-document Coreference Resolution | # Event Perfect | Number of event pairs with perfect match. |
| | # Event No Match | Number of event pairs with no match. |
| | # Entity Perfect | Number of entity pairs with perfect match. |
| | # Entity Reverse | Number of entity pairs with reverse match. |
| | # Entity No Match | Number of entity pairs with no match. |
| | # Perfectly Clustered + NEC | The number of pairs with NEC score of at least T and perfect clustering for event coreference resolution. |
| Connected Components | # Connected components | The number of connected components in G_{p_1, p_2} . |
| | Average component size | The average size of connected components in G_{p_1, p_2} . |
| Clique | # In Clique | The number of pairs in support-pairs(p_1, p_2) that are in a clique. |

Table 2: All 17 features used by our scorer, for a given predicate paraphrase pair p_1, p_2 , as detailed in Section 3.1.

| |
|--|
| [Police] ₀ [arrest] [two men] ₁ in incident at Westboro Beach. [Police] ₀ [kill] [man] ₁ in Vegas hospital who grabbed gun. |
| [Police] ₀ [arrest] [man] ₁ in incident at Westboro Beach. [Officers] ₀ [seize] [guy] ₁ in incident at Westboro Beach. |

Table 3: Examples of coreference errors made by Chirps and corrected by Barhom et al. (2019): 1) false positive: wrong man / two men alignment (disregarding location modifiers). 2) (hypothetical) false negative: lexically-divergent yet semantically-similar arguments.

Match otherwise. Also, we extract *Perfectly Clustered with NE Coverage* that combines both Named Entity coverage and coreference-resolution, which count the number of pairs their events are perfectly clustered and with NEC score of at least T .

Connected Components The original Chirps score of a predicate paraphrase pair is proportional to two parameters: (1) the number of supporting pairs; (2) the ratio of number of days in which

supporting pairs were matched relative to the entire collection period. The latter lowers the score of paraphrase pairs which might have been mistakenly aligned on relatively few days (e.g. due to misleading argument alignments in particular events). The number of days in which the predicates were aligned is taken as a proxy for the number of different events in which the predicates co-refer. Here, we aim to get a more reliable partition of tweets to different events by constructing a graph of tweets as nodes, with supporting tweet pairs as edges, and looking for connected components.

To that end, we define a bipartite graph $G_{p_1, p_2} = (V, E)$ for a candidate paraphrase pair, where $V = \text{tweets}(p_1, p_2)$ contains all the tweets in which p_1 or p_2 appeared, and $E = \text{support-pairs}(p_1, p_2)$. We compute C , the number of connected components in G_{p_1, p_2} , and define the following group: $\text{ConComp} = \{c \in C : |c| > 2\}$, which represents the number

of connected components with size greater than 2. From this group we derive two features $\#connected(p_1, p_2) = |ConComp|$ which represents the number of the connected components and $avg_connected(p_1, p_2)$, which is the average size of the connected components in the graph. A larger number of connected components indicates that the two predicates were aligned across a large number of likely different events.

Clique We similarly build a global tweet graph for all the predicate pairs, $G_{all} = (V', E')$, where $V' = \cup_{(p_1, p_2)} tweets(p_1, p_2)$, and $E' = \cup_{(p_1, p_2)} support_pairs(p_1, p_2)$. We compute Q , the set of cliques in G_{all} of size greater than 2. We assume that a pair of tweets are more likely to be coreferring if they are part of a bigger clique, whereas if they were extracted by mistake they wouldn't share many neighbors. We extract the following feature of clique coverage for a candidate paraphrase pair: $CLC(p_1, p_2) = |\{t_1^j, t_2^j \in support_pairs(p_1, p_2) : \exists q \in Q \text{ such that } t_1^j \in q \wedge t_2^j \in q\}|$.

3.2 Distantly Supervised Labels

In order to learn to score the paraphrases, we need gold standard labels, i.e., labels indicating whether a pair of predicate templates collected by Chirps is indeed a paraphrase. Instead of collecting manual annotations for a sample of the Chirps data, we chose a low-budget distant supervision approach. To that end, we leverage the similarity between the predicate paraphrase extraction and the event coreference resolution tasks, and use the annotations from the ECB+ dataset.

Our dataset consists of the predicate paraphrases from Chirps that appear in ECB+ (denoted ch-ECB+). As positive examples we consider all pairs of predicates p_1, p_2 from Chirps that appear in the same event cluster in ECB+, e.g., from $\{talk, say, tell, accord\ to, confirm\}$ we extract $(talk, say)$, $(talk, tell)$, ..., $(accord\ to, confirm)$.

Obtaining negative examples is a bit trickier. We consider as negative example candidates pairs of predicates p_1, p_2 from Chirps, which are *under the same topic*, but in different event clusters in ECB+, e.g., given the clusters $\{specify, reveal, say\}$, and $\{get\}$, we extract $(specify, get)$, $(reveal, get)$, and (say, get) .

Note that the ECB+ annotations are context-dependent. Thus a pair of predicates that are in principle coreferable may be annotated as non-

| | Pre Annotation | Post Annotation |
|-------------------------|----------------|-----------------|
| Train # positive | 266 | 803 |
| # negative | 2040 | 1056 |
| Dev # positive | 93 | 222 |
| # negative | 539 | 318 |
| Test # positive | 131 | 352 |
| # negative | 758 | 411 |

Table 4: Statistics of the paraphrase scorer dataset. The difference in size before and after the annotation is due to omitting examples with less than 3 supporting pairs.

| Model | ch-ECB+ Test Set | | | ch-Random |
|---------|------------------|-------------|-----------------------------|-------------|
| | AP | Accuracy | (P / R / F1) | AP |
| GloVe | 50.7 | 55.7 | (58.1 / 14.2 / 22.8) | 50.1 |
| Chirps | 62.5 | 60.4 | (59.9 / 45.7 / 51.6) | 51.4 |
| Chirps* | 80.0 | 73.8 | (74.1 / 66.5 / 70.1) | 59.5 |

Table 5: Ranker (AP) and classification (Accuracy, Precision, Recall, $F1$) results on ch-ECB+ test set (middle column), and Ranker results on ch-Random, a subset of 500 randomly selected predicate pairs from the entire Chirps resource (right column).

coreferring in a given context. To reduce the rate of such false-negative examples, we validated all the candidate negative examples and a sample of the positive examples using Amazon Mechanical Turk. Following [Shwartz et al. \(2017\)](#), we annotated the templates while presenting 3 argument instantiations from their original tweets. Thus, we only included in the final data predicate pairs with at least 3 supporting pairs. We required that workers have 99% approval rate on at least 1,000 prior tasks and pass a qualification test.

Each example was annotated by 3 workers. We aggregated the per-instantiation annotations using majority vote and considered a pair as positive if at least one instantiation was judged as positive. The data statistics are given in Table 4. This validation phase balanced the positive-negative proportion of instances in the data, from approximately 1:7 to approximately 4:5.

3.3 Model

We trained a random forest classifier ([Breiman, 2001](#)) implemented by the scikit-learn framework ([Pedregosa et al., 2011](#)). To tune the hyperparameters, we ran a 3 fold cross-validation randomized search, yielding the following values: 157 estimators, max depth of 8, minimum samples leaf of 1, and min samples split of 10.²

²We chose random forest over a neural model because of the small size of the training set.

| Rank | GloVe | Chirps | Chirps* |
|------|---------------------|--------------------|-------------------|
| 1 | take over / take up | announce / unveil | launch / unveil |
| 11 | begin / continue | introduce / unveil | buy / purchase |
| 21 | find / go | accuse / warn | add / introduce |
| 31 | move / plan | launch / unveil | announce / launch |
| 41 | buy / pay | announce / enter | accuse / warn |
| 51 | die / kill | accuse / threaten | cast / set |
| 61 | announce / confirm | arrest / charge | say / warn |
| 71 | deal / say | kill / tell | fire / use |
| 81 | condemn / deny | attack / strike | cause / trigger |
| 91 | believe / move | claim / kill | refresh / upgrade |

Table 6: Highly ranked paraphrases from the intersection of the candidate coreference pairs in the ECB+ test set and Chirps (ch-ECB+ test set). The table presents 10 highly ranked paraphrases, as ranked by each of GloVe, Chirps and our method, taken from ranks 1, 11, 21 and so on in each ranking. Pairs labeled as positive in our gold dataset are highlighted in purple.

| Ablated Feature Set | AP | Classification Metrics | | | |
|-----------------------|--------------|------------------------|--------------|--------------|----------------|
| | | Acc. | P | R | F ₁ |
| Chirps | 73.94 | 72.96 | 74.36 | 52.25 | 61.38 |
| Named Entity Coverage | 76.43 | 74.44 | 73.86 | 58.56 | 65.33 |
| Coreference | 75.22 | 72.04 | 72.32 | 51.8 | 60.37 |
| Connected Component | 76.59 | 73.51 | 74.53 | 54.05 | 62.66 |
| Clique | 76.78 | 73.19 | 72.51 | 55.85 | 63.1 |
| All Features | 77.13 | 73.88 | 73.96 | 56.3 | 63.94 |

Table 7: Ablation test results on the ch-ECB+ dev set for the ranking and classification models.

3.4 Evaluation

We used the model for two purposes: (1) classification: determining if a pair of predicate templates are paraphrases or not; and (2) ranking the pairs based on the predicted positive class score. We consider the ranking evaluation as more informative, as we expect the ranking to reflect the number of contexts in which a pair of predicates may be coreferring. That is, predicate pairs that are coreferring in many contexts will be ranked higher than those that are coreferring in just a few contexts.

We compare our model with two baselines: the original Chirps scores, and a baseline that assigns each pair of predicates the cosine similarity scores between the predicates using GloVe embeddings (Pennington et al., 2014).³ For the classification decisions made by the two baseline scores (Chirps score and cosine similarity for GloVe vectors), we learn a threshold that yields the best accuracy score over the train set, above which a pair of predicates is classified as positive.

³We were motivated to compare with GloVe since this resource is utilized as a lexical representation in the state-of-the-art system of (Barhom et al., 2019), which is utilized in the next section. Here, multi-word predicates were represented by the average of their GloVe word vectors.

Table 5 displays the accuracy, precision, recall and F_1 scores for classification evaluation and the Average Precision (AP) for ranking evaluation. Our scorer dramatically improves upon the baselines in all metrics.

To show that the improved scoring generalizes beyond examples that appear in the ECB+ dataset, we selected a random subset of 500 predicate pairs with at least 6 support pairs from the entire Chirps resource and annotated them in the same method described in Section 3.2. The ranker evaluated on this subset gained 8 points in AP, relative to the original Chirps ranking. All results are statistically significant using bootstrap and permutation tests with $p < 0.001$ (Dror et al., 2018).

Table 6 exemplifies highly ranked predicate pairs by our Chirps* scorer, the original Chirps scorer and the GloVe scorer, which illustrates the improved ranking performance of Chirps* (as measured in table 5 by the AP score).

Ablation Test To evaluate the importance of each type of feature, we perform an ablation test. Table 7 displays the performance of various ablated models, each of which with one set of features (Section 3.1) removed from the representation. In the classification task, removing the named entity coverage features somewhat improved the performance, mostly by increasing the recall. However, in terms of the (primary) ranking evaluation, each set of features contributed to the performance, with the full model performing best.

4 Leveraging a Paraphrasing Resource to Improve Coreference

In Section 3 we showed that leveraging CD event coreference annotations and model improves predicate paraphrase ranking. In this section, we show that this co-dependence can be used in both directions, and that using Chirps* as an external resource can improve the performance of a CD model.

As a preliminary analysis, we computed Chirps’ coverage of lexically-divergent pairs of co-referring event mentions in ECB+. We found approximately 30% coverage overall and above 50% coverage for coreferring verbal mentions.⁴ This indicates a substantial coverage of the lexically-divergent positive coreferrability decisions that need to be made in ECB+. In absolute numbers, Chirps covers

⁴Non-verbal mentions in ECB+ include nominalizations (*investigation*), names (*Oscars*) acronyms (*DUI*), and more. Chirps, by design, consists of verbal predicates only.

| Model | MUC | | | B ³ | | | CEAF- <i>e</i> | | | CoNLL |
|---|-------------|--------------|----------------|----------------|--------------|----------------|----------------|--------------|----------------|----------------|
| | R | P | F ₁ | R | P | F ₁ | R | P | F ₁ | F ₁ |
| LEMMA BASELINE | 76.5 | 79.9 | 78.1 | 71.7 | 85 | 77.8 | 75.5 | 71.7 | 73.6 | 76.5 |
| Barhom et al. (2019) | 77.6 | 84.5 | 80.9 | 76.1 | 85.1 | 80.3 | 81 | 73.8 | 77.3 | 79.5 |
| Barhom et al. (2019) + CHIRPS* FEATURES | 78.7 | 84.67 | 81.61 | 75.87 | 85.91 | 80.5 | 81.09 | 74.77 | 77.8 | 80.0 |

Table 8: Event mentions coreference-resolution results on ECB+ test set.

about 370 lexically-divergent pairs of coreferring event mentions appearing in the ECB+ training set, and about 200 in the test set.

4.1 Integration Method

The state-of-the-art CD coreference resolution model, by Barhom et al. (2019), trained a pairwise mention scoring function, $MLP_{scorer}(m_i, m_j)$, which predicts the probability that two mentions m_i, m_j refer to the same event. The mention representation includes a lexical component (GloVe embeddings) as well as a contextual component (ELMo embeddings, Peters et al., 2018). The mention pair representation $\vec{v}_{i,j}$, which is fed to the pairwise scorer, combines the two separate mention representations.

We extended the model by changing the input to the pairwise event mention scoring function to include information regarding the mention pair from Chirps*, as illustrated in Figure 1. We defined $\vec{v}_{i,j} = [\vec{v}_{i,j}; \vec{c}_{i,j}]$, where $\vec{c}_{i,j}$ denotes the Chirps* features, computed in the following way:

$$\vec{c}_{i,j} = \begin{cases} MLP_{ch}(\vec{f}_{m_i, m_j}) & \text{if } m_i, m_j \in \text{Chirps} \\ MLP_{ch}(\vec{0}) & \text{otherwise} \end{cases}$$

$\vec{f}_{m_i, m_j} \in \mathcal{R}^{17}$ is the feature vector representing a pair of predicates (m_i, m_j) for which there is an entry in Chirps, otherwise the input is a zero vector. MLP_{ch} is an MLP with a single hidden layer of size 50 and output layer of size 100, which is used to transform the discrete values in \vec{f}_{m_i, m_j} into the same embedding space of $\vec{v}_{i,j}$. The rest of the model remains the same, including the model architecture, training, and inference.⁵

4.2 Evaluation

We evaluate the event coreference performance on ECB+ using the official CoNLL scorer (Pradhan et al., 2014). The reported metrics are MUC (Vilain et al., 1995), B³ (Bagga and Baldwin, 1998),

⁵We also tried to incorporate only the final Chirps* score into the mention pair representation, but the performance improvement was smaller.

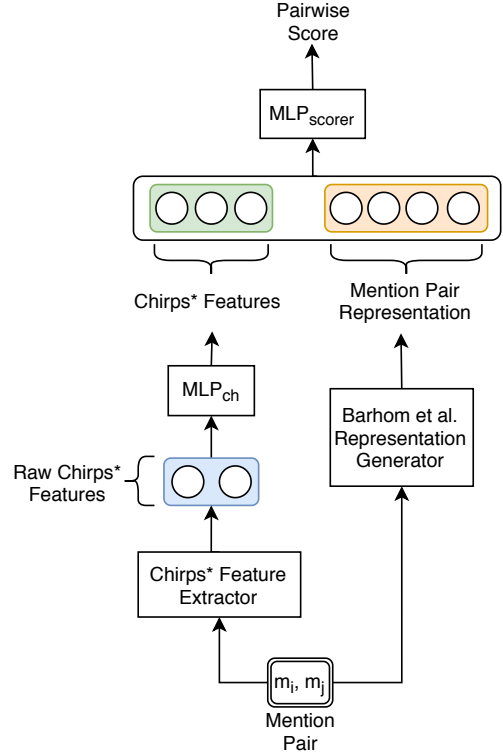


Figure 1: An illustration of the integrated mention pair scorer. The right vector is the original mention pair vector from Barhom et al. (2019), and the left one is our Chirps* extension, which is transformed through MLP_{ch} into the same embedding space. The two vectors are concatenated to form the mention pair representation, which is fed to the scoring function MLP_{scorer} .

CEAF-*e* (Luo, 2005) and CoNLL F_1 (the average of MUC, B³ and CEAF-*e* scores).

We compare the integrated model to the original model and to the lemma baseline which clusters together mentions that share the same mention-head lemma. The results in Table 8 show that the Chirps-enhanced model provides an improvement of 3.5 points over the lemma baseline and a small improvement upon Barhom et al. (2019) in all F_1 score measures. The greatest improvement is in the link-based MUC measure, which counts the number of corresponding links between the mentions. The Chirps component helps link more coreferring mentions (improving recall) and prevents the linking of some wrong mentions (improving precision).

Although the gap between our model and the

original model by Barhom et al. (2019) is statistically significant (bootstrap and permutation tests with $p < 0.001$), it is rather small. We can attribute it partly to the coverage of Chirps over ECB+ (around 30%), which entails that the majority of event mention pairs still have the same representation as in the original model. We also note that ECB+ suffers from annotation errors, as was observed by Barhom et al. (2019) and others.

5 Conclusion and Future Work

We studied the synergy between the tasks of identifying predicate paraphrases and event coreference resolution, both concerned with matching the meanings of lexically-divergent predicates, and showed that they can benefit each other. Using event coreference annotations as distant supervision, we learned to re-rank predicate paraphrases that were initially ranked heuristically, and managed to increase their average precision substantially. In the other direction, we incorporated knowledge from our re-ranked predicate paraphrases resource into a model for event coreference resolution, yielding a small improvement upon previous state-of-the-art results. We hope that our study will encourage future research to make further progress on both tasks jointly.

Acknowledgements

This work was supported in part by grants from Intel Labs, Facebook, the Israel Science Foundation grant 1951/17, the Israeli Ministry of Science and Technology and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

- Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Granada.
- Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. [Revisiting joint modeling of cross-document entity and event coreference resolution](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.
- Regina Barzilay and Lillian Lee. 2003. [Learning to paraphrase: An unsupervised approach using multiple-sequence alignment](#). In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 16–23.
- Regina Barzilay and Kathleen R. McKeown. 2001. [Extracting paraphrases from a parallel corpus](#). In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57, Toulouse, France. Association for Computational Linguistics.
- Cosmin Bejan and Sanda Harabagiu. 2010. [Unsupervised event coreference resolution with rich linguistic features](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Uppsala, Sweden. Association for Computational Linguistics.
- Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Unsupervised event coreference resolution. *Computational Linguistics*, 40(2):311–347.
- Jonathan Berant. 2012. *Global Learning of Textual Entailment Graphs*. Ph.D. thesis, Tel Aviv University.
- Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.
- Prafulla Kumar Choubey and Ruihong Huang. 2017. [Event coreference resolution by iteratively unfolding inter-dependencies among events](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133, Copenhagen, Denmark. Association for Computational Linguistics.
- Agata Cybulska and Piek T. J. M. Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *LREC*.
- Bill Dolan, Chris Brockett, and Chris Quirk. 2005. Microsoft research paraphrase corpus. *Retrieved March*, 29(2008):63.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. [The hitchhiker’s guide to testing statistical significance in natural language processing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.
- Matthew Honnibal and Ines Montani. 2017. *Spacy. Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*.

- Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. [Resolving event coreference with supervised representation learning and clustering-oriented regularization](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10, New Orleans, Louisiana. Association for Computational Linguistics.
- Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. [A continuously growing dataset of sentential paraphrases](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. [Joint entity and event coreference resolution across documents](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.
- Dekang Lin and Patrick Pantel. 2001. [DIRT - Discovery of inference rules from text](#). In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.
- Xiaoqiang Luo. 2005. [On coreference resolution performance metrics](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. [Paraphrasing revisited with neural machine translation](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. [Scoring coreference partitions of predicted mentions: A reference implementation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.
- Marta Recasens, Matthew Can, and Daniel Jurafsky. 2013. [Same referent, different words: Unsupervised mining of opaque coreferent mentions](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906, Atlanta, Georgia. Association for Computational Linguistics.
- Yusuke Shinyama and Satoshi Sekine. 2006. [Preemptive information extraction using unrestricted relation discovery](#). In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311, New York City, USA. Association for Computational Linguistics.
- Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of the second international conference on Human Language Technology Research*, pages 313–318. Morgan Kaufmann Publishers Inc.
- Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017. [Acquiring predicate paraphrases from news tweets](#). In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 155–160, Vancouver, Canada. Association for Computational Linguistics.
- Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48.
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics.
- Wei Xu, Alan Ritter, Chris Callison-Burch, William B Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.
- Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent bayesian model for

event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.

Congle Zhang and Daniel S Weld. 2013. Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786.