Resolving Event Coreference with Supervised Representation Learning and Clustering-Oriented Regularization

Kian Kenyon-Dean
School of Computer Science
McGill University
kian.kenyon-dean@mail.mcgill.ca

Jackie Chi Kit Cheung
School of Computer Science
McGill University
jcheung@cs.mcgill.ca

Doina Precup
School of Computer Science
McGill University
dprecup@cs.mcgill.ca

Abstract

We present an approach to event coreference resolution by developing a general framework for clustering that uses supervised representation learning. We propose a neural network architecture with novel Clustering-Oriented Regularization (CORE) terms in the objective function. These terms encourage the model to create embeddings of event mentions that are amenable to clustering. We then use agglomerative clustering on these embeddings to build event coreference chains. For both within- and cross-document coreference on the ECB+ corpus, our model obtains better results than models that require significantly more pre-annotated information. This work provides insight and motivating results for a new general approach to solving coreference and clustering problems with representation learning.

1 Introduction

Event coreference resolution is the task of determining which *event mentions* expressed in language refer to the same real-world event instances. The ability to resolve event coreference has improved the quality of downstream tasks such as automatic text summarization (Vanderwende et al., 2004), questioning-answering (Berant et al., 2014), headline generation (Sun et al., 2015), and text-mining in the medical domain (Ferracane et al., 2016).

Event mentions are comprised of an action component (or, head) and surrounding arguments. Consider the following passages, drawn from two different documents; the heads of the event mentions are in boldface and the subscripts indicate mention IDs:

(1) The president's **speech**$_{m1}$ **shocked**$_{m2}$ the audience. He **announced**$_{m3}$ several new controversial policies.

(2) The policies **proposed**$_{m4}$ by the president will not **surprise**$_{m5}$ those who **followed**$_{m6}$ his **campaign**$_{m7}$.

In this example, $m1$, $m3$, and $m4$ form a chain of coreferent event mentions (underlined), because they refer to the same real-world event in which the president gave a speech. The other four are singletons, meaning that they all refer to separate events and do not corefer with any other mention.

This work investigates how to learn useful representations of event mentions. Event mentions are complex objects, and both the event mention heads and the surrounding arguments are important for the event coreference resolution task. In our example above, the head words of mentions $m2$, *shocked*, and $m5$, *surprise*, are lexically similar, but the event mentions do not corefer. This task therefore necessitates a model that can capture the distributional relationships between event mentions and their surrounding contexts.

We hypothesize that prior knowledge about the task itself can be usefully encoded into the representation learning objective. For our task, this prior means that the embeddings of corefential event mentions should have similar embeddings to each other (a "natural clustering", using the terminology of Bengio et al. (2013)). With this prior, our model creates embeddings of event mentions that are directly conducive for the clustering task of building event coreference chains. This is contrary to the indirect methods of previous work that rely on pairwise decision making followed by a separate model that aggregates the sometimes inconsistent decisions into clusters (Section 2).

We demonstrate these points by proposing a method that learns to embed event mentions into a space that is tuned specifically for clustering. The representation learner is trained to predict which event cluster the event mention belongs to,

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 1–10
New Orleans, June 5-6, 2018.

using an hourglass-shaped neural network. We propose a mechanism to modulate this training by introducing *Clustering-Oriented Regularization* (CORE) terms into the objective function of the learner; these terms impel the model to produce similar embeddings for coreferential event mentions, and dissimilar embeddings otherwise.

Our model obtains strong results on within- and cross-document event coreference resolution, matching or outperforming the system of Cybulska and Vossen (2015) on the ECB+ corpus on all six evaluation measures. We achieve these gains despite the fact that our model requires significantly less pre-annotated or pre-detected information in terms of the internal event structure. Our model's improvements upon the baselines show that our supervised representation learning framework creates new embeddings that capture the abstract distributional relations between samples and their clusters, suggesting that our framework can be generalized to other clustering tasks[1].

2 Related Work

The recent work on event coreference can be categorized according to the assumed level of event representation. In the predicate-argument alignment paradigm (Roth and Frank, 2012; Wolfe et al., 2013), links are simply drawn between predicates in different documents. This work only considers cross-document event coreference (Wolfe et al., 2013, 2015), and no within-document coreference. At the other extreme, the ACE and ERE datasets annotate rich internal event structure, with specific taxonomies that describe the annotated events and their types (Linguistic Data Consortium, 2005, 2016). In these datasets, only within-document coreference is annotated.

The creators of the ECB (Bejan and Harabagiu, 2008) and ECB+ (Cybulska and Vossen, 2014), annotate events according to a level of abstraction between that of the predicate-argument approach and the ACE approach, being most similar to the TimeML paradigm (Pustejovsky et al., 2003). In these datasets, both within-document and cross-document coreference relations are annotated. We use the ECB+ corpus in our experiments because it solves the lack of lexical diversity found within the ECB by adding 502 new annotated documents, providing a total of 982 documents.

[1] All code used in this paper can be found here: https://github.com/kiankd/events

Previous work on model design for event coreference has focused on clustering over a linguistically rich set of features. Most models require a pairwise-prediction based supervised learning step which predicts whether or not a pair of event mentions is coreferential (Bagga and Baldwin, 1999; Chen et al., 2009; Cybulska and Vossen, 2015). Other work focuses on the clustering step itself, aggregating local pairwise decisions into clusters, for example by graph partitioning (Chen and Ji, 2009). There has also been work using non-parametric Bayesian clustering techniques (Bejan and Harabagiu, 2014; Yang et al., 2015), as well as other probabilistic models (Lu and Ng, 2017). Some recent work uses intuitions combining representation learning with clustering, but does not augment the loss function for the purpose of building clusterable representations (Krause et al., 2016; Choubey and Huang, 2017).

3 Event Coreference Resolution Model

We formulate the task of event coreference resolution as creating clusters of event mentions which refer to the same event. For the purposes of this work, we define an event mention to be a set of tokens that correspond to the *action* of some event. Consider the sentence below (borrowed from Cybulska and Vossen (2014)):

(3) On Monday Lindsay Lohan **checked into** rehab in Malibu, California after a car **crash**.

Our model would take, as input, feature vectors (see Section 4) extracted from the two event mentions (in bold) independently. In this paper, we use the gold-standard event mentions provided by the dataset, and leave mention detection to other work.

3.1 Model Overview

Our approach to resolving event coreference consists of the following steps:

1. Train a supervised neural network model which learns event mention embeddings by predicting the event cluster in the training set to which the mention belongs (Figure 1).

2. At test time, use the previously trained model's embedding layer to derive representations of unseen event mentions. Then, perform agglomerative clustering with these embeddings to create event coreference chains (Figure 2).

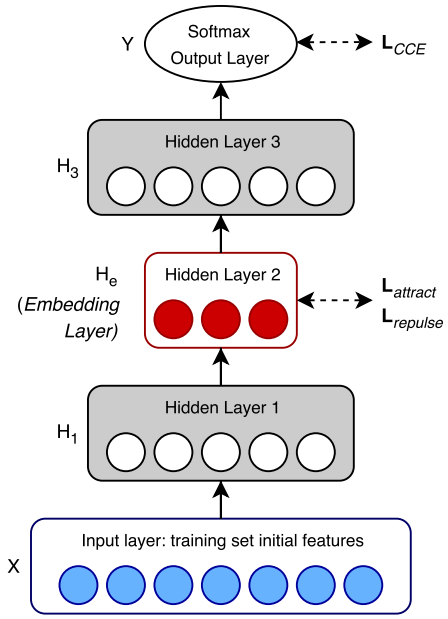

Figure 1: Our supervised representation learning model during the training step. Dashed arrows indicate contributions to the loss function.

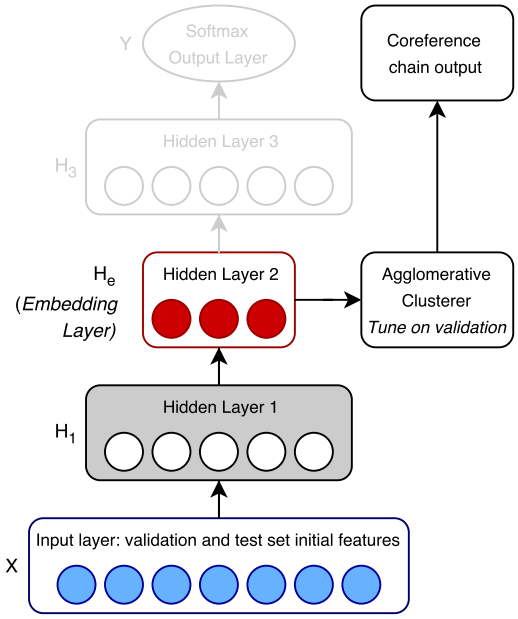

Figure 2: Our trained model at inference time, used for validation tuning and final testing. Note that H_3 and Y are not used in this step.

3.2 Supervised Representation Learning

We propose a representation learning framework based on training a multi-layer artificial neural network, with one layer H_e chosen to be the embedding layer. In the training set, there are a certain number of event mentions, each of which belongs to some gold standard cluster, making C total non-singleton clusters in the training set. The network is trained as if it were encountering a C+1-class classification problem, where the class of an event mention corresponds to a single output node, and all singleton mentions belong to class C+1[2].

When using this model to cluster a new set of mentions, the final layer's output will not be directly informative since the output node structure corresponds to the clusters within the training set. However, we hypothesize that the trained model will have learned to capture the abstract distributional relationships between event mentions and clusters in the intermediate layer H_e. We thus use the activations in H_e as the embedding of an event mention for the clustering step (see Figure 2). A similar hourglass-like neural architecture design has been successful in automatic speech recognition (Grézl et al., 2007; Gehring et al., 2013), but has not to our knowledge been used to pre-train embeddings for clustering.

[2]If each singleton mention (i.e., a mention that does not corefer with anything else) had its own class then the model would be confronted with a classification problem with thousands of classes, many of which would only have one sample; this is much too ill-posed, so we merge all singletons together during the training step.

3.3 Categorical-Cross-Entropy (CCE)

Using CCE as the loss function trains the model to correctly predict a training set mention's corresponding cluster. With model prediction y_{ic} as the probability that sample i belongs to class c, and indicator variable $t_{ic} = 1$ if sample i belongs to class c (else $t_{ic} = 0$), we have the mean categorical-cross entropy loss over a randomly sampled training input batch X:

$$\mathbf{L}_{CCE} = -\frac{1}{|\mathrm{X}|}\sum_{i=1}^{|\mathrm{X}|}\sum_{c=1}^{C+1} t_{ic}\log(y_{ic}) \qquad (1)$$

3.4 Clustering-Oriented Regularization (CORE)

With CCE, the model may overfit towards accurate prediction performance for those particular clusters found in the training set without learning an embedding that captures the nature of events in general. This therefore motivates introducing regularization terms based on the intuition that embeddings of mentions belonging to the same cluster should be similar, and that embeddings of mentions belonging to different clusters should be dissimilar. Accordingly, we define dissimilarity between two vector embeddings $(\vec{e_1}, \vec{e_2})$ according

to the cosine-distance function **d**:

$$\mathbf{d}(\vec{e_1}, \vec{e_2}) = \frac{1}{2}\Big(1 - \frac{\vec{e_1} \cdot \vec{e_2}}{||\vec{e_1}||||\vec{e_2}||}\Big) \quad (2)$$

Given input batch X, we create two sets $\mathcal{S}$ and $\mathcal{D}$, where $\mathcal{S}$ is the set of all pairs (a, b) of mentions in X that belong to the same cluster, and $\mathcal{D}$ is the set of all pairs (c, d) in X that belong to different clusters. Note that all vector embeddings $\vec{e_i} = \mathrm{H}_e(i)$; i.e., they are obtained by feeding the event mention i's features through to embedding layer H_e. We now define the following *Attractive* and *Repulsive* CORE terms.

3.4.1 Attractive Regularization

The first desirable property for the embeddings is that mentions that belong to the same cluster should have low cosine distance between each others' embeddings, since the agglomerative clustering algorithm uses cosine distance to make coreference decisions.

Formally, for all pairs of mentions a and b that belong to the same cluster, we would like to minimize the distance between their embeddings $\vec{e_a}$ and $\vec{e_b}$. We call this "attractive" regularization because we want to attract embeddings closer to each other by minimizing their distance $\mathbf{d}(\vec{e_a}, \vec{e_b})$ so that they will be as similar as possible.

$$\mathbf{L}_{attract} = \frac{1}{|\mathcal{S}|} \sum_{(a,b)\in\mathcal{S}} \mathbf{d}(\vec{e_a}, \vec{e_b}) \quad (3)$$

3.4.2 Repulsive Regularization

The second desirable property is that the embeddings corresponding to mentions that belong to different clusters should have high cosine distance between each other. Thus, for all pairs of mentions c and d that belong to different clusters, the goal is to maximize their distance $\mathbf{d}(\vec{e_c}, \vec{e_d})$. This is "repulsive" because we train the model to push away the embeddings from each other to be as distant as possible.

$$\mathbf{L}_{repulse} = 1 - \frac{1}{|\mathcal{D}|} \sum_{(c,d)\in\mathcal{D}} \mathbf{d}(\vec{e_c}, \vec{e_d}) \quad (4)$$

3.5 Loss Function

Equation 5 below shows the final loss function[3]. The attractive and repulsive terms are weighted by hyperparameter constants λ_1 and λ_2 respectively:

$$\mathbf{L} = \mathbf{L}_{CCE} + \lambda_1 \mathbf{L}_{attract} + \lambda_2 \mathbf{L}_{repulse} \quad (5)$$

By adding these regularization terms to the loss function, we hypothesize that the new embeddings of test set mentions (obtained by feeding-forward their features into the trained model) will exemplify the desired properties represented by the loss function, thus assisting the agglomerative clustering task in producing correct coreference-chains.

3.6 Agglomerative Clustering

Agglomerative clustering is a non-parametric "bottom-up" approach to hierarchical clustering, in which each sample starts as its own cluster, and at each step, the two most similar clusters are merged, where similarity between two clusters is measured according to some similarity metric. After each merge, clustering similarities are recomputed according to a preset criterion (e.g., single- or complete-linkage). In our models, clustering proceeds until a pre-determined similarity threshold, τ, is reached. We tuned τ on the validation set, doing grid search for $\tau \in [0, 1]$ to maximize B^3 accuracy[4]. Preliminary experimentation led us to use cosine-similarity (see cosine distance in Equation 2) to measure vector similarity, and single-linkage for clustering decisions.

We experimented with two initialization schemes for agglomerative clustering. In the first scheme, each event mention is initialized as its own cluster, as is standard. In the second, we initialized clusters using the lemma-δ baseline defined by Upadhyay et al. (2016). This baseline merges all event mentions with the same head lemma that are in documents with document-level similarity that is higher than a threshold δ. Upadhyay et al. showed that it is a strong indicator of event coreference, so we experimented with initializing our clustering algorithm in this way. We call this model variant CORE+CCE+LEMMA, and describe the parameter tuning procedures in more detail in Section 5.

4 Feature Extraction

We extract features that do not require the preprocessing step of event-template construction to represent the context (unlike Cybulska and Vossen

[3]Note that, while we present Equations 3 and 4 as summations over pairs from the input batch, the computation is actually reasonable when written in terms of matrix multiplications. The most expensive operation multiplying the embedded batch of input samples times its transpose.

[4]We optimize with B^3 F1-score because the other measures are either too expensive to compute (CEAF-M, CEAF-E, BLANC), or are less discriminative (MUC).

1.action	*checked into, crash*
2.time	*On Monday*
3.location	*rehab in Malibu, California*
4.participant	*Lindsay Lohan* (human) *car* (non-human)

Table 1: An event template of the sentence in Example 3, borrowed from Cybulska and Vossen (2014; 2015). Our model only requires as input the *action*, not the *time*, *location*, nor *participant* arguments.

(2015), see Table 1); instead, we represent the surrounding context by using the tokens in the general vicinity of the event's action. We thus only require the event's action – which is what we define as an *event mention* – to be previously detected, not all of its arguments. We motivate this by arguing that it would be preferable to build high quality coreference chains without event template features since since extracting event templates can be a difficult process, with the possibility of errors cascading into the event coreference step.

4.1 Contextual

Inspired by the approach of Clark and Manning (2016) in the entity coreference task, we extract, for the token sets below, (i) the token's *word2vec* word embedding (Mikolov et al., 2013) (or average if there are multiple); and, (ii) the one-hot count vector of the token's lemma[5] (or sum if there are multiple), for each event mention, em:

- the first token of em;
- the last token of em;
- all tokens in the em;
- each of the two tokens preceding em;
- each of the two tokens following em;
- all of the five tokens preceding em;
- all of the five tokens following em;
- all of the tokens in em's sentence.

4.2 Document

It is necessary to include features that characterize the mention's document, hoping that the model learns a latent understanding of relations between documents. We extract features from the event mention's document by building lemma-based TF-IDF vector representations of the document. We use log normalization of the raw term frequency of token lemma t in document d, $f_{t,d}$, where $TF_t = 1 + \log(f_{t,d})$. For the IDF term we use smoothed inverse document frequency, with N as the number of documents and n_t as the number of documents that contain the lemma, we have $IDF_t = \log(1 + \frac{N}{n_t})$. By performing a component-wise multiplication of the *IDF* vector with each row in term-frequency matrix *TF*, we create TF-IDF vectors of each document in the training and test sets (with length corresponding to the number of unique lemmas in the training set). We compress these vectors to 100 dimensions with principal component analysis fitted onto the train set document vectors, which is used to transform the validation and test set document vectors.

[5]This is a 500-dimensional vector where the first 499 entries correspond to the 499 most frequently occurring lemmas in the training set, and the 500^{th} entry indicates if the lemma is not in that set of most frequently occurring lemmas.

4.3 Comparative

We include comparative features to relate a mention to the other mentions in its document and to the mentions in the set of documents the model would be requested to extract event coreference chains from. This is motivated by the fact that coreference decisions must be informed by the relationship mentions have with each other. Firstly, we encode the position of the mention in its document with specific binary features indicating if it is first or last; for example, if there were five mentions and it were the third, this feature would correspond to the vector $[0, \frac{3}{5}, 0]$.

Next, we define two sets of mentions we would like to compare with: the first contains all mentions in the same document as the current mention em, and the second contains all mentions in the data we are asked to cluster. For each of these sets, we compute: the average word overlap and average lemma overlap (measured by harmonic similarity) between em and each of the other mentions in the set. We thus add two feature vector entries for each of the sets: the average word overlap between em and the other mentions in the set, and the average lemma overlap between em and the other mentions in the set.

5 Experimental Design

We run our experiments on the ECB+ corpus, the largest corpus that contains both within- and cross-document event coreference annotations. We followed the train/test split of Cybulska and Vossen (2015), using topics 1-35 as the train set and 36-45 as the test set. During training, we split off a

validation set[6] for hyperparameter tuning.

Following Cybulska and Vossen, we used the portion of the corpus that has been manually reviewed and checked for correctness. Some previous work (Yang et al., 2015; Upadhyay et al., 2016; Choubey and Huang, 2017) do not appear to have followed this guideline from the corpus creators, as they report different corpus statistics compared to those reported by Cybulska and Vossen. As a result, those papers may report results on a data set with known annotation errors.

5.1 Evaluation Measures

Since there is no consensus in the coreference resolution literature on the best evaluation measure, we present results obtained according to six different measures, as is common in previous work. We use the scorer presented by Pradhan et al. (2014). In this task, the term "coreference chain" is synonymous with "cluster".

MUC (Vilain et al., 1995). Link-level measure which counts the minimum number of link changes required to obtain the correct clustering from the predictions; it does not account for correctly predicted singletons.

B^3 (Bagga and Baldwin, 1998). Mention-level measure which computes precision and recall for each individual mention, overcoming the singleton problem of MUC, but can problematically count the same coreference chain multiple times.

CEAF-M (Luo, 2005). Mention-level measure which reflects the percentage of mentions that are in the correct coreference chains. Note that precision and recall are the same in this measure since we use pre-annotated mentions.

CEAF-E (Luo, 2005). Entity-level measure computed by aligning predicted with the gold chains, not allowing one chain to have more than one alignment, overcoming the problem of B^3.

BLANC (Luo et al., 2014). Computes two F-scores in terms of the pairwise quality of coreference decisions and non-coreference decisions, and averages these scores together for the final results.

CoNLL. The mean of MUC, B^3, and CEAF-E.

5.2 Models

We compare our representation-learning model variants to three baselines: a deterministic lemma-based baseline, a lemma-δ baseline, and an unsupervised baseline which clusters the originally extracted features. We also compare with the results of Cybulska and Vossen (2015).

5.2.1 Baselines

LEMMA. This algorithm clusters event mentions which share the same head word lemma into the same coreference chains across all documents.

LEMMA-δ. Proposed by Upadhyay et al. (2016), this method provides a difficult baseline to beat. A δ-similarity threshold is introduced, and we merge two mentions with the same head-lemma if and only if the cosine-similarity between the TF-IDF vectors of their corresponding documents is greater than δ. This δ parameter is tuned to maximize B^3 performance on the validation set, which we found occurs when $\delta = 0.67$.

UNSUPERVISED. This is the result obtained by agglomerative clustering over the original unweighted features. Again, we optimize the τ similarity threshold over the validation set.

5.2.2 Sentence Templates (CV2015)

Cybulska and Vossen (2015) propose a model that uses sentence-level event templates (see Table 1), requiring more annotated information than our models. See (Vossen and Cybulska, 2017) for further elaboration of this model. To our knowledge, this is the best previous model on ECB+ using the same data and evaluation criteria as our work.

5.2.3 Representation Learning.

We test four different model variants:

- CCE: uses only categorical-cross-entropy in the loss function (Equation 1);
- CORE: uses only clustering-oriented regularization; i.e., the attract and repulse terms (Equations 3 and 4);
- CORE+CCE: includes categorical-cross-entropy and the attract and repulse terms (Equation 5);
- CORE+CCE+LEMMA: initializes the agglomerative clustering with clusters computed by lemma-δ (with a differently tuned value of δ than the baseline) and continues the clustering process using the similarities between the embeddings created by CORE+CCE.

[6] Topics 2, 5, 12, 18, 21, 23, 34, 35 (randomly chosen).

Model	λ_1	λ_2	B^3	τ
Baselines				
UNSUPERVISED	-	-	0.590	0.657
LEMMA	-	-	0.597	-
LEMMA-δ	-	-	0.612	-
Model Variants				
CORE+CCE+L	2.0	0.0	**0.678**	0.843
CORE+CCE	2.0	2.0	0.663	0.776
	2.0	1.0	0.666	0.773
	2.0	0.1	0.665	0.843
	2.0	0.0	**0.669**	0.843
	0.0	2.0	0.662	0.710
CORE	2.0	2.0	0.631	0.701
	1.0	1.0	0.625	0.689
CCE	-	-	0.644	0.853

Table 2: Model comparison based on validation set B^3 accuracy with optimized τ cluster-similarity threshold. For CORE+CCE+LEMMA (indicated as CORE+CCE+L) we tuned to $\delta = 0.89$; for LEMMA-δ we tuned to $\delta = 0.67$.

5.3 Hyper-parameter Tuning

For the representation learning models, we performed a non-exhaustive hyper-parameter search optimized for validation set performance. We keep the following parameters constant across the model variants:

- 1000 neurons in H_1 and H_3; 250 neurons in H_e, the embedding layer (see Figure 1);
- Softmax output layer with $C + 1$ units;
- ReLU activation functions for all neurons;
- *Adam* gradient descent (Kingma and Ba, 2014);
- 25% dropout between each layer;
- Learning rate of 0.00085 (times 10^{-1} for CORE);
- Randomly sampled batches of 272 mentions, where a batch is forced to contain pairs of coreferential and non-coreferential mentions.

Models are trained for 100 epochs. At each epoch, we optimize τ (our agglomerative clustering similarity threshold) using a two-pass approach: we first test 20 different settings of τ, then τ is further optimized around the best value from the first pass. For CORE+CCE+LEMMA, we tune the δ parameter of the lemma-δ clustering approach to the validation set by testing 100 different values of δ; these different δ values initialize the clusters, and we then continue clustering by testing validation results obtained when using the similarities between the embeddings created by CORE+CCE for different values of τ.

Some of the results of hyperparameter tuning on the validation set are shown in Table 2. Interestingly, we observe that CORE+CCE performs slightly better with $\lambda_2 = 0$; i.e., without repulsive regularization. This suggests that enforcing representation similarity is more important than enforcing division, although we cannot conclusively state that repulsive regularization would not be useful for other tasks. Nonetheless, for test set results we use the optimal hyperparameter configurations found during this validation-tuning step; e.g., for CORE+CCE we set $\lambda_1 = 2$ and $\lambda_2 = 0$.

6 Results

Table 3 presents the performance of the models for combined within- and cross-document event coreference. Results for these models are obtained with the hyper-parameter settings that achieved optimal accuracy during validation-tuning.

Firstly, we observe that CORE+CCE offers marked improvements upon the UNSUPERVISED baseline, CORE model, and CCE model. From these results we conclude: (i) supervised representation learning provides more informative embeddings than the original feature vectors; and, (ii) that combining Clustering-Oriented Regularization with categorical-cross-entropy is better than just using one or the other, indicating that our introduction of these novel terms into the loss function is a useful contribution.

We also note that CORE+CCE+LEMMA (which obtains the best validation set results) beats the strong LEMMA-δ baseline. Our model offers marked improvements or roughly equivalent scores in each evaluation measure except BLANC, where the baseline offers a 3 point F-score improvement. This is due to the very high precision of the baseline, whereas CORE+CCE+LEMMA seems to trade precision for recall.

We finally observe that CORE+CCE+LEMMA improves upon the results of Cybulska and Vossen (2015). We obtain improvements of 14 points in MUC, 3 points in entity-based CEAF, 5 points in CoNLL, and 1 point in BLANC, with equivalent results in B^3 and mention-based CEAF. These re-

	MUC			B^3			CM	CE			BLANC			CoNLL
Model	R	P	F	R	P	F	F	R	P	F	R	P	F	F
Baselines														
LEMMA	66	58	62	66	58	62	51	87	39	54	64	61	63	61
LEMMA-δ	55	68	61	61	80	**69**	**59**	73	60	66	62	80	**67**	66
UNSUPERVISED	39	63	48	55	81	66	51	72	49	58	57	58	58	57
Previous Work														
CV2015	43	77	55	58	86	**69**	58	-	-	66	60	69	63	64
Model Variants														
CCE	66	63	65	69	60	64	50	59	63	61	69	56	59	63
CORE	58	58	58	66	58	62	44	53	53	53	66	54	56	57
CORE+CCE	62	70	66	67	69	68	56	73	64	68	68	59	62	67
CORE+CCE+LEMMA	67	71	**69**	71	67	**69**	58	71	67	**69**	72	60	64	**69**

Table 3: Combined within- and cross-document test set results on ECB+. Measures CM and CE stand for mention-based CEAF and entity-based CEAF, respectively.

	MUC			B^3			CM	CE			BLANC			CoNLL
Model	R	P	F	R	P	F	F	R	P	F	R	P	F	F
Baselines														
LEMMA-δ	41	77	53	86	97	**92**	85	92	82	87	65	86	71	77
UNSUPERVISED	32	36	34	85	86	85	74	80	78	79	65	55	57	66
Model Variants														
CCE	44	49	46	87	89	88	79	82	80	81	67	67	67	72
CORE	55	32	40	89	70	78	65	64	79	71	75	54	56	63
CORE+CCE	43	68	53	87	95	91	84	90	82	86	67	76	70	76
CORE+CCE+LEMMA	57	69	**63**	90	94	**92**	**86**	90	86	**88**	73	78	**75**	**81**

Table 4: Within-document test set results on ECB+. Note that LEMMA is equivalent to LEMMA-δ in the within-document setting. Cybulska and Vossen (2015) did not report the performance of their model in this setting.

sults suggest that high quality coreference chains can be built without necessitating event templates.

In Table 4, we see the performance of our models on within-document coreference resolution in isolation. These results are obtained by cutting all links drawn across documents for the gold standard chains and the predicted chains. We observe that, across all models, scores on the mention- and entity-based measures are substantially higher than the link-based measures (e.g., MUC and BLANC). The usefulness of CORE+CCE+LEMMA (which initializes the clustering with the lemma-δ predictions and then continues to cluster with CORE+CCE) is exemplified by the improvements or matches in every measure when compared to both LEMMA-δ and CORE+CCE. The most vivid improvement here is observed with the 10 point improvement in MUC over both models as well as the 4 and 5 point improvements in BLANC respectively, where the higher recall entails that CORE+CCE+LEMMA confidently predicts coreference links that would otherwise have been false negatives.

7 Conclusions and Future Work

We have presented a novel approach to event coreference resolution by combining supervised representation learning with non-parametric clustering. We train an hourglass-shaped neural network to learn how to represent event mentions in a useful way for an agglomerative clustering algorithm. By adding the novel Clustering-Oriented Regularization (CORE) terms into the loss function, the model learns to construct embeddings that are easily clusterable; i.e., the prior that embeddings of samples belonging to the same cluster should be similar, and those of samples belonging to different clusters should be dissimilar.

Our results suggest that clustering embeddings created with representation learning is much better than clustering of the original feature vectors, when using the same agglomerative clustering algorithm. We show that including CORE in the loss function improves performance more than when only using categorical-cross-entropy to train the representation learner model. Our top-performing model obtains results that improve upon previous work despite the fact that our model requires less annotated information in order to perform the task.

Future work involves applying our model to automatically annotated event mentions and other event coreference datasets, and extending this framework toward a full end-to-end system that does not rely on manual feature engineering at the input level. Additionally, our model may be useful for other clustering tasks, such as entity coreference and document clustering. Lastly, we seek to determine how CORE and its imposition of a clusterable latent space structure may or may not assist in improving the quality of latent representations in general.

Acknowledgements

This work was funded with grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche du Québec - Nature et Technologies (FRQNT). We thank the anonymous reviewers for their helpful comments and suggestions.

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.

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*, pages 1–8. ACL.

Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Unsupervised Event Coreference Resolution. *Computational Linguistics*, 40(2):311–347.

Cosmin Adrian Bejan and Sanda M Harabagiu. 2008. A Linguistic Resource for Discovering Event Structures and Resolving Event Coreference. In *LREC*.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. 2014. Modeling Biological Processes for Reading Comprehension. In *Proceedings of the 2014 Conference on EMNLP*, pages 1499–1510.

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for NLP*, pages 54–57. ACL.

Zheng Chen, Heng Ji, and Robert Haralick. 2009. A Pairwise Event Coreference Model, Feature Impact and Evaluation for Event Coreference Resolution. In *Proceedings of the workshop on events in emerging text types*, pages 17–22. ACL.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. *arXiv preprint arXiv:1707.07344*.

Kevin Clark and Christopher D Manning. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. *arXiv preprint arXiv:1606.01323*.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *LREC*, pages 4545–4552.

Agata Cybulska and Piek Vossen. 2015. Translating Granularity of Event Slots into Features for Event Coreference Resolution. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 1–10.

Elisa Ferracane, Iain Marshall, Byron C Wallace, and Katrin Erk. 2016. Leveraging coreference to identify arms in medical abstracts: An experimental study. *EMNLP*, pages 86–95.

Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel. 2013. Extracting deep bottleneck features using stacked auto-encoders. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3377–3381. IEEE.

Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky. 2007. Probabilistic and bottle-neck features for LVCSR of meetings. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–757. IEEE.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. 2016. Event linking with sentential features from convolutional neural networks. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 239–249.

Linguistic Data Consortium. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. Version 5.4.3 2005.07.01.

Linguistic Data Consortium. 2016. Rich ERE Annotation Guidelines Overview. V4.2.

Jing Lu and Vincent Ng. 2017. Learning antecedent structures for event coreference resolution. In *Machine Learning and Applications (ICMLA), 2017*

16th IEEE International Conference on, pages 113–118. IEEE.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and EMNLP*, pages 25–32. ACL.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard H Hovy. 2014. An Extension of BLANC to System Mentions. In *ACL (2)*, pages 24–29.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard H Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *ACL (2)*, pages 30–35.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Michael Roth and Anette Frank. 2012. Aligning predicate argument structures in monolingual comparable texts: A new corpus for a new task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 218–227. ACL.

Rui Sun, Yue Zhang, Meishan Zhang, and Donghong Ji. 2015. Event-driven headline generation. In *Proceedings of ACL*, pages 462–472.

Shyam Upadhyay, Nitish Gupta, Christos Christodoulopoulos, and Dan Roth. 2016. Revisiting the Evaluation for Cross Document Event Coreference. In *COLING*.

Lucy Vanderwende, Michele Banko, and Arul Menezes. 2004. Event-centric summary generation. *Working notes of DUC*, pages 127–132.

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. ACL.

Piek Vossen and Agata Cybulska. 2017. Identity and granularity of events in text. *arXiv preprint arXiv:1704.04259*.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2015. Predicate argument alignment using a global coherence model. In *Proceedings of the 2015 Conference of NAACL: Human Language Technologies*, pages 11–20.

Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu, et al. 2013. PARMA: A Predicate Argument Aligner. In *ACL (2)*, pages 63–68.

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.