Cross-Document Coreference Resolution for Entities and Events

Ph.D. Thesis Proposal

(draft as of April 19, 5pm)

Chris Tanner

May 16, 2018

Abstract of “Cross-Document Coreference Resolution for Entities and Events”

Coreference Resolution is a fundamental natural language processing (NLP) problem, as it attempts to resolve which underlying discourse objects refer to one another. Further, it serves as an essential component of many other core NLP tasks, including information extraction, question-answering, document summarization, etc. However, decades of research have primarily focused on resolving *entities* (people, locations, organizations), with significantly less attention given to *events* – the actions performed. Also, systems almost always use third-party software to first determine *which* exact pieces of text (i.e., “mentions”) to resolve, and these two lines of research have remained disjoint.

This proposal outlines research which aims to improve coreference resolution by taking a comprehensive approach. First, we develop a novel state-of-the-art event-based system which (1) uses significantly fewer lexical features than most existing systems; and (2) overcomes pitfalls from the commonly-used clustering approaches. Next, we plan to jointly model both entities and events, while uniquely leveraging each model to benefit the other. Last, we propose merging the mention detection and coreference lines of research, with the idea that accurately performing coreference on a given set of candidate mentions could improve mention detection, and vice versa.

Contents

1 Introduction 3
1.1 Motivation . . . 3
1.2 Problem Statement . . . 3
1.3 Coreference Systems . . . 5
1.3.1 Mention Detection . . . 5
1.3.2 Coreference Resolution . . . 5

2 Related 7
2.1 Motivation . . . 7
2.2 Deep Learning Approaches . . . 7
2.2.1 Coreference Resolution for Entities . . . 7
2.2.2 Systems using ECB+ Corpus . . . 8

3 Corpora 9
3.1 Event Coreference Corpora . . . 9
3.1.1 ECB: EventCorefBank . . . 9
3.1.2 EECB: Extended ECB . . . 9
3.1.3 ECB+: EventCorefBank+ . . . 9

4 Event Coreference Resolution 11
4.1 Initial Approaches . . . 11
4.2 CCNN: Conjoined Convolutional Neural Network . . . 12
4.2.1 Input Features . . . 12
4.2.2 Architecture . . . 13
4.2.3 Loss / Optimization . . . 14
4.3 Neural Clustering (NC) . . . 14
4.3.1 Existing Clustering Approaches . . . 14
4.3.2 Our Approach . . . 14
4.3.3 Architecture . . . 15
4.3.4 Inference . . . 15
4.4 Our Coreference Systems . . . 15
4.4.1 Training / Development / Testing Data . . . 15
4.4.2 Within-Document . . . 16
4.4.3 Cross-Document . . . 16
4.5 Results . . . 16
4.5.1 Comparison to Others' Systems . . . 17
4.5.2 Error Analysis . . . 18

5 Proposed Work 19
5.1 Joint Entity and Event Coreference . . . 19
5.2 Motivation . . . 19
5.3 Related Work . . . 19
5.4 Approach . . . 19

5.4.1 Semantic Trees 20
5.4.2 Joint Work 20
5.5 Comprehensive Coreference: Mention Detection + Coreference 20
5.6 Motivation 20
5.7 Related Work 21
5.8 Approach 21

Bibliography **22**

Chapter 1

Introduction

Thesis Statement: I propose a novel, neural-based mention-pair model for cross-document coreference resolution for events, which uses few lexical features and addresses shortcomings of traditional clustering approaches. I will extend this work by jointly modelling both entities and events, while using structured information (e.g, parse trees). Last, we aim to improve mention detection, whereby we develop an all-inclusive, end-to-end system which jointly resolves mention boundaries and coreference predictions.

m_i

m_j, where

$m_j \in \{m_1, m_2, ..., m_{i-1}, \epsilon\}$

$\oplus \sum\limits_{w_i \in M_{post}} emb(lemma(w_i))$

1.1 Motivation

Coreference Resolution remains a fundamental NLP task, as it is an essential component for any system that desires "understanding" textual data. That is, in order to accurately model meaning, one must at the very least understand which items are concerning the same underlying objects. As a simple example, if one performs a Web search for "President Barack Obama", some of the web page search results will contain sentences which only refer to him as "Obama", "he", or "The President," and correctly using this information is essential for returning relevant information to the user's query. Further, coreference resolution is useful for information extraction [22], question answering [32], topic detection [1], summarization [12], and more.

1.2 Problem Statement

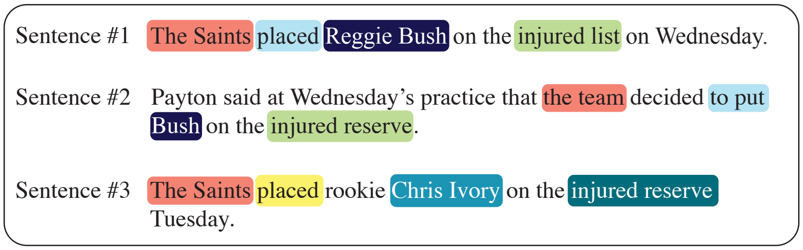
Coreference resolution is the task of identifying – within a single text or across multiple documents – which *mentions* refer to the same underlying discourse object.

A **mention** is a particular instance of word(s) in a document which represent an *entity* or *event*, such as *Barack Obama*, *he*, or *announced*.

An **entity** may be a person, location, time, or an organization. The mentions which refer to them may be *named*, *nominal*, or *pronominal*:

- Named mentions are represented by proper names (e.g., André Benjamin or Pakse, Laos)
- Pronominal mentions are represented by pronouns (e.g., she or it)
- Nominal mentions are represented by descriptive words, not composed entirely of a named entity or pronouns (e.g., The well-spoken citizen)

An **event** can generally be thought of as a specific action. Quine [36] was the first to propose that an event refers to a physical object which is grounded to a specific time and location, and that two events are identical (i.e., co-referent) if they share the same spatiotemporal location. This definition has become the general consensus within the community[1]. Specifically, two co-referent events must share the same *properties* and *participants*. For example, in Figure 1.1, sentences #1 and #2 contain the co-referent events ("placed" and "put"), yet neither are co-referent with events in sentence #3. Often times, the participants (arguments) may be referred to in different ways, implied, or missing altogether.



Sentence #1 The Saints placed Reggie Bush on the injured list on Wednesday.

Sentence #2 Payton said at Wednesday's practice that the team decided to put Bush on the injured reserve.

Sentence #3 The Saints placed rookie Chris Ivory on the injured reserve Tuesday.

Figure 1.1: Sample of a coreference resolution corpus (ECB+), depicting gold coref mentions as having shared box colors.

Coreference resolution is concerned with linking either entities together and/or events together; that is, entities shall not be linked to events, and doing so would be considered an incorrect link. Although one may be interested in evaluating coreference systems by their ability to correctly link *pairs* of mentions [45], coreference resolution is ultimately a clustering task, whereby we wish to group all like-mentions together, as shown with colored boxes in Figure 1.1. Specifically, coreference systems aim to find a globally-optimal fit of mentions to clusters, whereby every mention m in the corpus is assigned to exactly one cluster C, such that every $m_i, m_j \in C$ are co-referent with each other. If a given m_i is not anaphoric with any other m_j, then it should belong to its own C with a membership of one.

Given a corpus of text documents, coreference resolution can be performed and evaluated on either a **within-document** or **cross-document** basis:

- **Within-document** is when each mention may only link to either (1) no other mention; or (2) other mentions which are contained in the same document. Even if the gold truth data denotes a mention should link with a mention from a different document, we ignore these links during the evaluation.
- **Cross-document** is when the entire corpus is available for linking; a mention is eligible to be co-referent with mentions in any other document, and the evaluation reflects the same. As described in [42], cross-document evaluation is normally conducted by transforming the entire corpus into a "meta-document."

[1] Hovy, et. al. [21] provide an in-depth study of varying definitions.

1.3 Coreference Systems

Coreference systems are predicated upon first having entity/event mentions identified, via a separate, distinct task called *mention detection*. Next, these identified mentions are used by coreference resolution models.

1.3.1 Mention Detection

This initial mention identification process is a separate line of research and has remained a fundamental task of NLP for several decades [31]. When concerned with entities, research is commonly referred to as *named entity recognition* or *entity recognition*. When concerned with events, research is commonly referred to as *event detection*.

Named Entity Recognition

The earliest work started in 1991 with the task of identifying company names [38]. In 1996, the MUC-6 conference [18] focused on Information Extraction tasks, which included coining the phrase "named entity" and drastically increasing attention to mention detection. Early work demonstrated state-of-the-art performance with Hidden Markov Models (HMMs) [5] and Conditional Random Fields (CRFs) [30]. Presently, the best performing systems use similar models — but from a deep learning framework — which include Bi-directional LSTMs [7, 20] and Convolution Neural Nets (CNNs) [28].

Event Detection

Event Detection has received significantly less attention than Named Entity Recognition; however, the task of semantic role labelling (SRL) addresses a similar and more encompassing problem; SRL is a shallow semantic parsing task, whereby the goal is to identify each predicate in a sentence, along with its constituents and how they fill a semantic role — specifically, to determine the role (e.g., Agent, Patient, Instrument, etc) and their adjuncts (Locative, Temporal, Manner, etc). [16, 35]. In short, both SRL systems and Event Detection systems have often relied on using many lexical and syntactical features, including those from constituency parsers [41], dependency parsers [23], etc. However, like entity recognition, recent state-of-the-art systems for Event Detection use Bi-directional LSTMs and CNNs [14].

1.3.2 Coreference Resolution

As mentioned, coreference systems aim to create the correct clusters of mentions; however, due to the number of possible mention-to-cluster combinations, finding a globally-optimal assignment of clusters is NP-Hard and thus computationally intractable. In attempt to avoid this, systems typically perform pairwise-mention predictions, then use those predictions to build clusters. The specific modelling strategies for such approximately fall into two categories: (1) mention-ranking / mention-pairs; and (2) entity-level / event-level, as described below; however, it is worth noting that there is no unanimously-dominate modelling paradigm, as state-of-the-art results often come from any of the ones listed below.

Mention-ranking models define a scoring function $f(m_i, m_j)$ which operates on a mention m_j and possible antecedent m_i, where m_i occurs earlier in the document and could be null (represented by ϵ and denoting that m_j is non-anaphoric); e.g., Wiseman, et. al.'s [44]. These models aim to find the ideal m_i antecedent for every m_j mention. After every mention has decided to link to ϵ or a previous mention, it is common practice to define each cluster simply by joining together all mentions which are connected by a single path. This is a potential weakness, as it asserts the transitive property holds true (e.g., if m_3 predicts m_2 as its antecedent, and m_2 predicts m_1 as its antecedent, then $\{m_1, m2, m3\}$ are all connected, which could be a bad decision, as there is no direct consideration given to the relatedness between m_1 and m_3.

Mention-pair models score all pairs (m_i, m_j), in contrast to mention-ranking models which aim to find the ideal m_i antecedent for every m_j. After every pair of mentions has been scored, it is common practice to cluster mentions in a best-first or easy-first manner (e.g. agglomerative clustering). Because mention-pair models base their predictions on the information from just two mentions at a time, they are by definition less expressive than entity/event-level models. Yet, their inference can be relatively simple and effective, allowing them to be fast and scalable. Consequently, they have often been the approach used by many state-of-the-art systems [13,40], including our work described in this proposal.

Entity/Event-level Instead of operating on a mention-level basis, these models differ in that they focus on building a global representation of each underlying entity or event, the basis of which determines each mention's membership [10,43]. These models are attractive due to the intuitive nature of modelling each entity with its own representation; however, challenges include (1) deciding how to represent each entity as it is being developed; (2) decided how many entities to model.

The aim is for the above definitions and descriptions to provide sufficient background to understand all of our subsequent chapters in this proposal, including related research, our research thus far, and our proposed work of (1) combining entity and event coreference into one model; and (2) combining our own in-house mention detection with our coreference models.

Chapter 2

Related

2.1 Motivation

The seminal research on event coreference can be traced back to the DARPA-initiated MUC conferences, whereby the focus was on limited scenarios involving terrorist attacks, plane crashes, management succession, resignation, etc. [2, 22].

In the present day, Deep Learning is revolutionizing NLP. And, although coreference resolution has been researched for several decades, only recently have a few publications successfully applied deep learning to coreference – almost all of which have been for *entity* coreference. We attribute this relatively small amount of deep learning models to the fact that coreference resolution is inherently a clustering task, which tends to be a non-obvious modality for deep learning. Since our work so far (1) uses deep learning and (2) operates on events by using the ECB+ corpus, we organize the related research accordingly.

2.2 Deep Learning Approaches

To the best of our knowledge, there are six publications which apply deep learning to coreference resolution, five of which focus on entity coreference (the sixth is for events and is listed in Section 2.2.2).

2.2.1 Coreference Resolution for Entities

Sam Wiseman, et. al. [43, 44] trained a mention-ranking model with a heuristic loss function that assigns different costs based on the types of errors made, and their latter work used mention-ranking predictions towards an entity-level model.

Clark and Manning [9, 10] also built both a mention-ranking model and an entity-level model, the former of which was novel in using reinforcement learning to find the optimal loss values for the same four distinct error types defined in Wiseman's, et. al. [44] work.

Most recently, Lee, et. al. [26] developed the first end-to-end coreference system which is only trained on gold clusters and uses few features (speaker information, genre, span distance, mention width) to do both mention detection and coreference resolution on those mentions. Notably, this paper is the most similar to our proposed, desired goal as described in Section 5.

2.2.2 Systems using ECB+ Corpus

For our research, we make use of the ECB+ corpus [11], which we further describe in Section 3. This rich corpus provides annotations for both entities and events, yet most research focuses on using *either* events *or* entities, not both. To the best of our knowledge, there are only two papers which focus on the event mentions of ECB+: The Hierarchical Distance-dependent Chinese Restaurant Process (HDDCRP) model by Yang, et. al. [46] (not a Deep Learning approach) and Choubey's and Huang's Iteratively-Unfolding approach [8] (a Deep Learning approach).

HDDCRP Model

Yang, et. al's HDDCRP model [46] uses a clever mention-pair approach, whereby they first use logistic regression to train parameters θ for the similarity function in Equation 2.1.

$$f_\theta(x_i, x_j) \propto \text{exp}\{\theta^T \psi(m_i, m_j)\} \tag{2.1}$$

Then, in a Chinese-restaurant-process fashion, they probabilistically link together mentions based purely on the scores provided by this similarity function. That is, the value of $f(m_i, m_j)$ is directly correlated with the probability of (m_i, m_j) being chosen as a linked pair. Then, identical to Bengtson's and Roth's work [4], the HDDCRP model forms clusters by tracing through all linked pairs. All mentions that are reachable by a continuous path become assigned the same cluster. This hinges on the transitive property of coreference. For example, if (m_1, m_3), (m_3, m_5) and (m_5, m_6) are each individually linked via the scoring function, then a cluster C_i is formed, where $C_i = \{m_1, m_3, m_5, m_6\}$, even though (m_1, m_5) or (m_3, m_6) may have had very low similarity scores. We aim to improve this shortcoming, as detailed in Section 4.3.

Neural Iteratively-Unfolding Model

Recently, Choubey and Huang [8] introduced the first neural model for event coreference. Their system also fits into the mention-pair paradigm, whereby mentions are predicted by a feed-forward network. The authors asserted that when using the ECB+ corpus, within-doc coreference did not benefit from using mention context, which is an important finding. However, similar to the weakness of the HDDCRP model, they merge clusters which contain *any* mention-pair whose predicted score is below a given threshold, independent of mentions' relation to the cluster at large.

Chapter 3

Corpora

Annotating a corpus with coreference information is an expensive task, as every sentence should include at least one entity and event. Since many of these mentions will refer to other mentions in the same sentence, document, or other documents, the task quickly becomes complex and time-consuming. Further, many mentions can be tricky to perfectly annotate, and consequently, annotators will often differ in their markings. Therefore, multiple annotators label every sentence, allowing a majority vote to resolve discrepancies. Due to these difficulties, there are not many event corpora, yet the ECB+ is sufficient for research, and we now describe its evolution.

3.1 Event Coreference Corpora

3.1.1 ECB: EventCorefBank

Created by Bejan and Harabagiu in 2010 [3], the ECB corpus provides within-document and cross-document event coreference annotations for 480 documents, spanning 43 disjoint *topics*. The documents were selected from GoogleNews archive[1] and each topic is a collection of documents (roughly 7-15 relatively short documents) which all concern the same seminal event, such as a particular arrest, transaction, attack, sporting event, election, etc. Each event mention, and its relation with another event, was annotated, where an even relation is one of six types: subevent, reason, purpose, enablement, precendence, and related. The weakness of this corpus is that (1) only a subset of the sentences were annotated; (2) only events were annotated – no entities.

3.1.2 EECB: Extended ECB

Lee, et. al. [25] extended the ECB corpus by addressing both of the aforementioned weaknesses; four annotators fully annotated all sentences, with entity coreference relations included. Also, they removed the originally-annotated relations, only keeping the coreference ones (e.g., subevent, purpose, related). Note, *light verbs* were not annotated (e.g., *make* an offer or *give* a talk).

3.1.3 ECB+: EventCorefBank+

All of our event-based experiments were conducted on this corpus, as it is the largest and commonly used corpus. It includes the 480 documents from the original ECB corpus, along with 502 additional documents which stem from 43

[1] http://news.google.com

	Train	Dev	Test	Total
# Documents	462	73	447	982
# Sentences	7,294	649	7,867	15,810
# Mentions-1	1,938	386	2,837	5,161
# Mentions-2	142	52	240	434
# Mentions-3	18	–	25	43
# Mentions-4	6	–	7	13

Table 3.1: Statistics of the ECB+ Corpus, where Mentions-N represents event mentions which are N-tokens in length.

additional topics which are highly similar to the original 43 topics from the original ECB corpus. For example, Topic 1 in the ECB corpus contains documents which concern Tara Reid checking into a rehab center in Malibu, California. However, Topic 1 in the ECB+ corpus also includes documents which concern Lindsay Lohan checking into a rehab center in Rancho Mirage, California. The purpose for this similarity is to help create a more realistic and potentially confusable scenario for the cross-document task. However, it has been shown [46] that one can simply perform document classification as a pre-processing step, which will allow a cross-document model to appropriately, perfectly confine itself to a relevant document set from which to consider linking mentions (e.g., ECB's Topic 1 documents can be easily separable from ECB+'s added Topic 1 documents).

We maintain the same train/dev/test splits as previous researchers, as further detailed in Chapter 4.4. A sample of the corpus in shown in Figure 1.1, and statistics are listed in Table 3.1, where it is clear that the majority of gold mentions are one token in length (e.g, *announced*). **NOTE:** the corpus contains 15,812 sentences. The corpus creators only place stock in their annotations for 1,840 specific sentences. However, they also annotate some additional sentences which they place less stock in (e.g., not all annotators labelled these additional sentences). All research papers which use this corpus simply evaluate again all labelled mentions, even if they do not belong to the well-supported 1,840 sentences. For fair comparison, we follow suit.

Chapter 4

Event Coreference Resolution

Our current work has primarily focused on event coreference resolution.

4.1 Initial Approaches

Our initial approaches were unsuccessful but largely informative: we tried modelling all mentions (cross-document) at the same time, in a *mention-pair* manner. Given N mentions in a given topic, we evaluated $\frac{N*(N-1)}{2}$ mention pairs, with the aim of predicting which were coref or not (i.e., a 0 or 1 prediction). Approximately 1 out of every 15 mention pairs are actually co-referenced (gold truth). Our supervised, classification modelling attempts included using various LSTM and SVM approaches.

LSTMs: The LSTM approaches were based on the fact that if we let m_{i-1} represent the word immediately before mention m_i in the corpus, then m_{i-1} will have high likelihood of predicting m_i. Our idea was that if m_{i-1} has high likelihood of also predicting mention m_j, then m_j and m_i might be coreference. Our premise was that the likelihood of predicting each mention is directly correlated with each mention's likelihood of referring to the same event as m_i. Despite many attempts, this never gave good results. My conclusions were that:

- LSTMs are sensitive and are too close to memorizing the exact sequence of words, especially given the variance in our corpus
- Our corpus' context varies too much and is not conducive to our premise. For example, one sentence may be "Barack Obama *spoke* ... " and another sentence may be "The President, on Tuesday, *spoke* ...". The word Tuesday is not likely to predict the word *spoke*.

SVMs: Our SVM approach used a variety of commonly-used lexical features. The performance exceeded the *SameLemma* baseline. However, difficulties included the class imbalance (too many negatives examples per positive example), along with the computational expense of training a proper kernel – yielding the model intractable.

We also tried a generative, **topic modelling** approach, a la LDA with Gibbs sampling. The underlying event was our latent variable: P(Event|Document) and P(Mention|Event). Again, our performance barely rivalled the baseline of *SameLemma*. My conclusion was that predicting the number of latent classes (traditionally, number of topics) is too difficult and drastically affects the performance, especially since nearly half of all events only have 1 mention.

From these attempts, along with the two previously published works which have used a ECB+ corpus [8,46], I drew the following conclusions:

- an event-level model is not ideal, since it is inappropriate to predict how many unique events exist
- negative sub-sampling is critical
- most importantly, predicting within-document links first is critical and easier, as there are fewer mentions, and it serves as a stepping stone before considering links across all documents. Then, one can use these predicted within-document clusters to merge with other within-doc clusters that reside in other documents.

4.2 CCNN: Conjoined Convolutional Neural Network

Conjoined Neural Networks (a.k.a. Siamese Networks) were first introduced by Bromly and LeCun [6] for the task of determining if two signatures were from the same person or not. Specifically, Conjoined Networks are two identical neural networks, each of which accepts distinct inputs, which are joined by a single loss function over their highest-level features. The loss function computes a similarity score (e.g., euclidean distance) for an input pair. The networks are said to be conjoined because they share the same weights and thus work together as one network that learns how to discriminate. The benefits of tying the weights are that it:

1. Ensures that similar inputs will be mapped appropriately, otherwise, they could be mapped to hidden representations that are dissimilar from their input representations; and
2. Forces the network to be symmetric. If we abstractly represent the Conjoined Network as a function, then: $CCNN(f_i, f_j) \equiv CCNN(f_j, f_i)$. This is critical, as the CCNN's similarity function should be independent of the ordering of its input pair.

Last, Conjoined Networks have been shown to perform well in low-resource situation [17]. This is ideal for our task, as it is highly likely that at test time we encounter event mentions that are out-of-vocabulary (OOV). We desire our model to discriminately learn the relationships of input mentions, rather than exclusively relying on the input values themselves. Likewise, we choose to use a Convolutional Network due to:

1. their power in learning sub-regions of features and the relations thereof, and
2. their recent advances in many NLP tasks [15, 24, 47].

4.2.1 Input Features

Since our CCNN needs each mention to be represented exclusively by its own input, we used none of the relational features[1] that are common in other coreference systems (e.g., SameLemma, Jaccard similarity of mentions' context, shared WordNet parents). We used Stanford CoreNLP [29] to extract the following features, which we thoroughly tested in different ways:

- **Part-of-Speech:** LSTM-learned POS embeddings; and 1-hot representations.
- **Lemmatization**: Lemmatized each token and represented it by pre-trained GloVe [33] word embeddings.
- **Dependency Lemma:** we represent the dependent parent/children of each token via their aforementioned lemma embeddings.

[1] We also experimented with extending our CCNN model by adding relational features as a merged-layer at the highest neural level.

- **Character Embeddings:** each token is represented as a concatenation of its character embeddings.
- **Word Embeddings:** pre-trained GloVe word embeddings.

We account for mentions' having varying token lengths by summing their tokens in place, thus representing each mention as a fixed-length vector.

4.2.2 Architecture

We define the full embedding for a given token t as $t_{emb} = t_{f_1} \oplus t_{f_2} \oplus \ldots \oplus t_{f_n}$, where $\oplus$ represents vector concatenation and t_{f_i} represents a specific input feature vector.

Naturally, we may want to convolve over the context of mention m, too, by including the N words before and after m. Thus, our input for mention m is a matrix M, and a la Kim [24], we zero-pad unfilled windows.

Let **M** represent the full matrix corresponding to mention m: $\mathbf{M} \in \mathbb{R}^{(2N+1)\times d}$ and $\mathbf{M}_{(i,j),(k:l)}$ represent the sub-matrix of M from (i, j) to (k, l).

We define a kernel with dimensions (h, w), where $h < (2N + 1)$ and $w < d$. This allows the kernel to operate on sub-sections of the embeddings. The kernel has an associated weight matrix $\mathbf{w} \in \mathbb{R}^{h\times w}$. Starting at a given index (i, j) within mention matrix **M**, a feature c_i is defined as:

$$c_i = f(\mathbf{w}^T \mathbf{M}_{(i:i+h-1),(j:j+w-1)} + b) \tag{4.1}$$

where $b \in \mathbb{R}$ is an added bias term. The kernel runs over every possible sub-section of mention matrix **M**, yielding a feature map $\mathbf{c} \in \mathbb{R}^{(2N-h)\times(d-w-1)}$

Since the network is comprised of two identical, conjoined halves, we sufficiently represent the architecture in Figure 4.1 with just one half. The Lambda function calculates the Euclidean distance of each half's univariate vector and emits a two-class softmax prediction regarding the likelihood of the two mentions being co-referent.

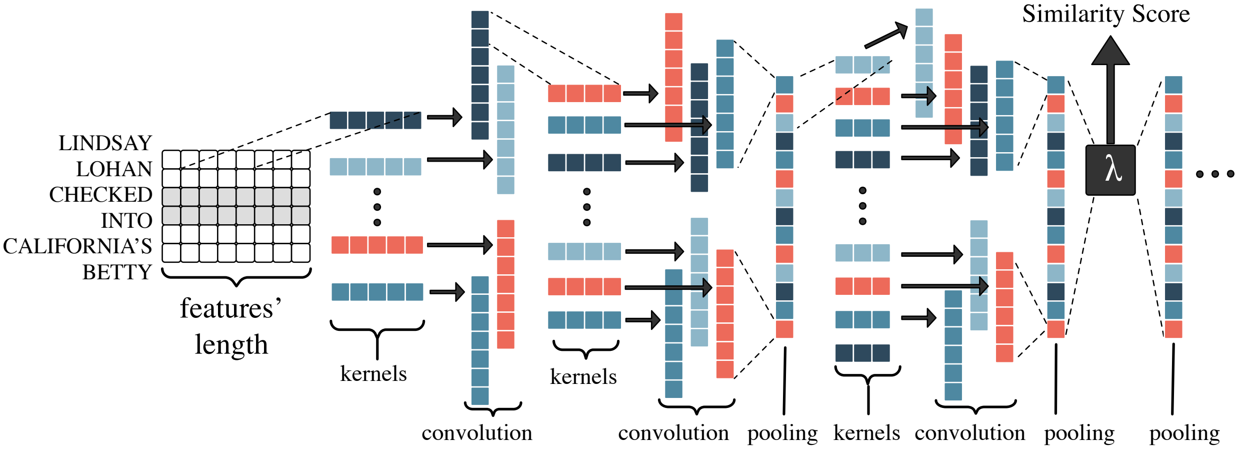


Figure 4.1: One half of the Conjoined Convolutional Neural Network's Architecture.

4.2.3 Loss / Optimization

Our goal is to maximize discriminability of different event mentions, while enforcing features to be as similar as possible when they are of the same event. Contrastive Loss, shown in Equation 4.2, is perfectly suited for this objective [27, 39]. Our training set has a strong class imbalance (most input pairs are not co-referent), so we down-sample to a 5:1 ratio of negative-to-positive examples. We use Adagrad for optimization.

$$L(\hat{y}, y) = \frac{1}{2N}\sum_{n=1}^{N}[(y)d^2 + (1-y) * (max(1-d, 0))^2] \\ \text{where } d = \|y_n - \hat{y}_n\|_2 \tag{4.2}$$

4.3 Neural Clustering (NC)

It is common practice for mention-pair models to first assign a probability score to every mention-pair, and then cluster with a different model.

4.3.1 Existing Clustering Approaches

Agglomerative Clustering is a simple but effective approach. It first assigns each mention to its own singleton cluster. Then, it repeatedly merges the two distinct clusters which contain the shortest-distance mention pairs. Although this is a strong baseline, as seen in Yang, et. al. [46], there are three main weaknesses:

1. One must define a stopping threshold α.
2. Any given α hinges on the data being uniform across documents. In reality, distances between mention-pairs could vary significantly between documents and topics.
3. Most significant, each cluster merge is based solely on two individual mentions, yet these mentions may not be representative of the cluster at large.

HDDCRP and Iterative-Folding Clustering both contain the issue #3 from above, as detailed in Sections 2.2.2 and 2.2.2.

4.3.2 Our Approach

We aim to use the strengths of agglomerative clustering, while replacing its shortcomings. We train a classifier to learn the most likely positive cluster merge, where the cluster is represented more holistically than a mention-pair basis.

Specifically, we learn a function $f(C_x, C_y)$ that predicts the likelihood of an appropriate, positive merging of clusters (C_x, C_y). Let $d(m_i, m_j)$ be the mention-pair distance predicted by our CCNN model, where $m_i \in C_x$, and $m_j \in C_y$. Function $f(C_x, C_y)$ is based on four simple features:

- min-pair distance: $\min_{m_i,m_j} d(m_i, m_j)$
- avg-pair distance: $\frac{\sum_{m_i,m_j} d(m_i,m_j)}{\|C_x\|\|C_y\|}$
- max-pair distance: $\max_{m_i,m_j} d(m_i, m_j)$
- size of candidate cluster: $\frac{\|C_x\|+\|C_y\|}{\sum_z \|C_z\|}$

The first three features serve to better represent the cluster at large (issue #3 from above). For example, a given cluster C_1, when evaluated against two other candidate clusters C_2 and C_3, may have the same minimum mention-pair distance score with both C_2 and C_3 Yet, the average and maximum distance scores shed more light onto which cluster has more similar mentions. Cluster size represents the size percentage of our considered merge, relative to all mentions in our current set. This may help prevent clusters from growing too large, and is not as vulnerable to issue #2.

4.3.3 Architecture

We define f as a feed-forward neural network[2] which predicts the probability of a positive cluster merge, via a two-class softmax function. Our loss function is weighted binary cross-entropy, to account for the class imbalance situation that most pairs of clusters should not be merged together.

4.3.4 Inference

Our system will incrementally build up clusters, starting with each cluster having just one mention (in the within-document scenario). Thus, it is important to train our Neural Clustering model on positive and negative examples of clusters in varying states of completeness. Our gold truth data informs us which mentions are co-referent, but since there is no single canonical ordering in which mentions should become co-referent, we generate synthetic data to represent possible positive and negative examples of when clusters should be merged.

Specifically, for training, we generate a positive example by randomly sampling a golden cluster, followed by splitting the cluster into two random subsets. The above four features are calculated for these two subsets of clusters, and the target output is a positive case. Likewise, we generate negative examples by sampling random subsets from disjoint golden clusters.

At test time, we use Neural Cluster to evaluate every possible (C_x, C_y) cluster pair in an easy-first manner. That is, at each iteration, we merge only the (C_x, C_y) pair that yielded the highest likelihood of a positive merge. Then, we re-evaluate all pairs with our newly merged cluster, and repeat until the model no longer predicts merge. Thus, unlike aforementioned models, we do not *require* additional stopping parameters.

4.4 Our Coreference Systems

We use our CCNN and Neural Clustering (NC) models together to perform coreference resolution. The only differences between the within-document and cross-document scenarios are our data and evaluation metric, as described below.

4.4.1 Training / Development / Testing Data

We adhere to the same data splits as previous researchers, whereby the dev set is topics 23-25, and the test set is topics 26-45. Traditionally, topics 1-22 are used as training. However, since our NC model relies on our CCNN's predictions, we remove topics 19-22 from the training set and instead use them as dev sets for our NC models. The complete details are provided in our Supplemental Materials document.

[2] We used 1 hidden layer of 25 units, ReLU activation without dropout, and Adagrad as our optimizer.

4.4.2 Within-Document

We train a CCNN model on mention-pairs which appear in the same document, and using its predictions on a held-out set, we train the NC to predict when to merge clusters.

4.4.3 Cross-Document

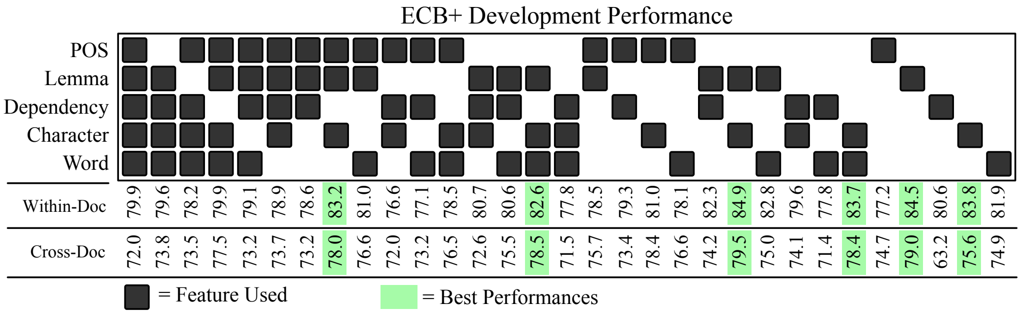
Cross-document resolution is a superset of the within-document task; it uses all coreference chains, regardless if mentions in a cluster were originally from the same document or not. Our cross-document and within-document systems are identical, except: (1) we train a separate CCNN only on mention-pairs which are from different documents; (2) instead of initializing our clustering with all singleton clusters, we use our within-document NC predictions as starting clusters; (3) at each iteration, we only consider merging clusters (C_x, C_y) if C_x and C_y contain mentions from disjoint sets of documents. Our cross-document NC only uses cross-document mention pairs distances for its decisions. Thus, cross-document merging will never merge two within-document clusters from the same document.

4.5 Results

As a recap, our research concerns three independent axis of investigation:

- **Features:** which features are most useful, and can we use few features?
- **Mention-Pair Model:** how well does CCNN perform against a standard feed-forward neural network[3] (FFNN)?
- **Clustering:** can we outperform Agglomerative via our Neural Clustering model?

Our metric is CoNLL F1 score, which is a clustering-based metric that combines the F1 scores of MUC, B^3, and $CEAF_e$, and we use the official scorer script (v8.01) [34].



ECB+ Development Performance

POS	■		■	■	■	■	■	■	■	■	■	■					■	■	■	■							■				
Lemma	■	■		■	■	■	■	■	■				■	■	■		■				■	■	■					■			
Dependency	■	■	■		■	■	■			■	■		■	■		■		■			■			■	■				■		
Character	■	■	■	■		■		■		■		■	■		■	■			■			■		■		■				■	
Word	■	■	■	■	■				■		■	■		■	■	■				■			■		■	■					■
Within-Doc	79.9	79.6	78.2	79.9	79.1	78.9	78.6	🟩 83.2	81.0	76.6	77.1	78.5	80.7	80.6	🟩 82.6	77.8	78.5	79.3	81.0	78.1	82.3	🟩 84.9	82.8	79.6	77.8	🟩 83.7	77.2	🟩 84.5	80.6	🟩 83.8	81.9
Cross-Doc	72.0	73.8	73.5	77.5	73.2	73.7	73.2	🟩 78.0	76.6	72.0	73.2	76.5	72.6	75.5	🟩 78.5	71.5	75.7	73.4	78.4	76.6	74.2	🟩 79.5	75.0	74.1	71.4	🟩 78.4	74.7	🟩 79.0	63.2	🟩 75.6	74.9

■ = Feature Used 🟩 = Best Performances

Figure 4.2: The CoNLL F1 performance of our flagship CCNN + Neural Clustering system, using all combinations of features.

We were interested in the five common, non-relational embedding features which are detailed in Section 4.2.1: POS, Lemma, Dependency Lemma, Character, Word. We tested all combinations of features on the Dev Set, and Lemma + Character Embedding yielded the best dev results (see Figure 4.2). Thus, our CCNN + Neural Clustering system used only these two features in its evaluation against other systems (see Table 4.1).

Our immediate evaluation measures the pairwise mention predictions. As shown in Table **??** [TODO: make table], our CCNN model ourperforms the FFNN, SVM, and SameLemma baselines.

[3] Given two mentions i and j, with corresponding feature vectors f_i and f_j, their input to the Feed-Forward Neural Network is the vector $\|f_i - f_j\|$

The final coreference results show that our CCNN model outperforms a FFNN, and that our Neural Clustering outperforms Agglomerative Clustering. Figure **??** shows an example of where the aforementioned weaknesses of Agglomerative are improved by NC. [TODO: add figure demonstrating AGG VS NC examples.] Further, when training and testing on gold mentions, we achieved CoNLL F1 scores of **81.2** and **72.4** for within-document and cross-document, respectively. We denote these scores as the new baseline to which to compare future systems.

	Within-Document				Cross-Document			
	MUC	B^3	CEAF	CoNLL F1	MUC	B^3	CEAF	CoNLL F1
Test Set: ECB+ Gold Mentions								
SameLemma	58.3	83.0	75.9	72.4	84.2	68.2	48.0	66.8
FFNN+AGG	59.9	85.6	78.4	74.6	77.7	69.9	50.1	65.9
FFNN+NC	60.7	86.7	79.4	75.6	74.9	67.8	56.3	67.0
CCNN+AGG	70.5	89.1	83.5	81.0	84.1	70.7	55.5	70.1
CCNN+NC	70.9	88.9	83.6	**81.2**	86.4	71.7	59.1	**72.4**
Test Set: HDDCRP's Predicted Mentions								
SameLemma	40.4	66.4	66.2	57.7	66.7	51.4	46.2	54.8
HDDCRP	53.4	75.4	71.7	66.8	73.1	53.5	49.5	58.7
CCNN+NC	54.0	75.5	72.2	**67.2**	71.3	57.0	49.6	**59.3**
Test Set: Choubey's et. al. Mentions								
SameLemma	48.8	66.7	65.1	60.2	68.1	53.3	47.2	56.2
Choubey	62.6	72.4	71.8	68.9	73.4	80.4	56.5	63.6
CCNN+NC	67.3	73.0	69.5	**69.9**	77.0	56.3	60.2	**64.5**

Table 4.1: Comparison against other systems, while our models use only the Lemma + Character Embedding features. FFNN denotes a Feed-Forward Neural Network Mention-Pair model. AGG denotes Agglomerative Clustering.

Lemma Embeddings were the most useful feature, followed closely by Character Embeddings. Since *SameLemma* has historically been a strong baseline, this is unsurprising.

Character Embeddings were effective for the same reason String Edit Distance is often a strong feature: there tends to be a direct correlation between the textual similarity of mentions and their likelihood of being co-referent. Both random character embeddings and pre-trained ones yielded the same performance, suggesting that the power comes from the uniqueness of characters, not any *meaning* conveyed in the characters.

Empirically, Lemma + Character Embeddings are complementary features; the semantic information conveyed within the lemma embeddings complement the syntactic information of character embeddings. Related, POS by itself was a poor feature, but combining it with either Lemma or Character Embeddings offered strong results.

Ideally, a classifier should learn how to combine all features such that the unhelpful ones are given no weight. However, in practice, that is often extremely difficult, due to both the large parameter space and high entropy wherein some combinations of features seem to equally help as much as hurt. Thus, we conclude that one should try to use the fewest features as possible for coreference resolution, then expand appropriately.

In all experiments, our results were inversely correlated with the amount of context our CCNN used. That is, our best performance came when we used no context, only the mention words. This agrees with the Choubey's, et. al. findings [8].

4.5.1 Comparison to Others' Systems

SameLemma has historically proven to be a strong baseline. That is, anytime two mentions have identical lemmas, simply mark them as being co-referent.

Using the same mentions that were used by the HDDCRP and Choubey (Iteratively-Unfolding) systems, our flagship CCNN+NC system yielded the highest results, despite using few features.

4.5.2 Error Analysis

False Positives include mentions which are textually identical but not actually co-referent (e.g., *placed* and *placed*, which only differ in their direct objects). **False Negatives** include abbreviations (e.g., *i.r.* and *injured reserve*) and colloquial phrases (e.g., The **casting** of Smith; (2) Smith **stepped into** the role; (3) was **handed the keys**).

[TODO: show actual examples]

Since our cross-document system relies on the within-document predictions, it is easy for errors to propagate.

Chapter 5

Proposed Work

5.1 Joint Entity and Event Coreference

5.2 Motivation

As shown in Figure [TODO: INSERT FIGURE], two sentences which contain co-referring entity mentions may also contain co-referring event mentions in a parallel fashion. Table [TODO: CALCULATE STATS] demonstrates how often this occurs in the ECB+ corpus. Since evidence of co-referring events increases the likelihood that the entities should also co-refer, we are motivated to model both entities and events, and to allow each model to influence the other.

5.3 Related Work

There has been some research which demonstrates benefits of jointly resolving mentions across multiple entities [TODO: cite the papers (p490 of Jurafsky)]. However, there has not been much research that uses event information to resolve entities. Haghighi and Klein [19] include a feature which concerns the governor of the head of nominal mentions (which could be events). Rahman and Ng [37] uses the semantic roles of entity mentions, along with the verb pairs of their predicates. These models use event information to help inform entity coreference; yet, they do not perform event coreference or use resolved events to inform entity coreference.

Choubey, et. al. [8] performs event coreference resolution via a feed-forward neural network. Afterwards, in an ad hoc fashion, their system [TODO: describe their system]

Most similar to our proposed work, Lee, et. al. [25] [TODO: describe their system and mention that they use their own corpus, EECB, not ECB+]

5.4 Approach

We aim to first use strong entity coreference system. We will evaluate both (1) our CCNN approach; and (2) Stanford Core NLP Toolkit's software on 3 different entity-labelled corpora:

- CoNLL-2012 Shared Task
- EECB (the corpus developed in [25])
- ECB+

5.4.1 Semantic Trees

Hopefully our CCNN approach outperforms Stanford's. If it is reasonably close, we will work on refining our model. Alternatively, I am interested in exploring semantic Tree-Based approaches, such as Tree-LSTM, and modelling the likelihood that two mentions' co-referring based on the similarity of their semantic trees. That is, one could learn common mappings that occur, such as the example in Figure **??** [TODO: make a figure to show two dependency trees]. A simple baseline could be Tree-Edit distance. More involved approaches include:

- seq2seq model (where the Tree is expanded to linear form)
- CNN model which learns patterns of sub-regions of trees
- ensemble of auto-encoders, each of which calculates the cost of mapping from one sub-region to another ($||f(Tree(m_1)) - f(Tree(m_2)))||$)

5.4.2 Joint Work

We will use whichever model above that gives the best results on the 3 corpora. Here, the emphasis of our research is not so much on developing the best possible entity coreference system, but to research the potential benefits from the joint modelling with events and to build each up in an iterative fashion. Our goal is to use a Expectation-Maximization (EM)-style approach. As a rough sketch (the math needs work and is not correct as is), we will aim for a back-and-forth like equations **??** and **??**.

$$\begin{aligned} P(m_{ent1}|m_{ent2}) &= \frac{Q(m_{ent1}|m_{ent2}) + P(m_{event1}|m_{event2})}{\sum\limits_{m_{enti}} [Q(m_{enti}|m_{ent2}) + P(m_{eventi}|m_{event2})]} \\ \text{where } \; Q(m_{ent1}|m_{ent2}) &= \frac{CCNN(m_{ent1}|m_{ent2})}{\sum\limits_{m_{enti}} CCNN(m_{enti}|m_{ent2})} \end{aligned} \tag{5.1}$$

$$\begin{aligned} P(m_{event1}|m_{event2}) &= \frac{Q(m_{event1}|m_{event2}) + P(m_{ent1}|m_{ent2})}{\sum\limits_{m_{eventi}} [Q(m_{eventi}|m_{event2}) + P(m_{enti}|m_{ent2})]} \\ \text{where } \; Q(m_{event1}|m_{event2}) &= \frac{CCNN(m_{event1}|m_{event2})}{\sum\limits_{m_{eventi}} CCNN(m_{eventi}|m_{event2})} \end{aligned} \tag{5.2}$$

5.5 Comprehensive Coreference: Mention Detection + Coreference

5.6 Motivation

Currently, Mention Detection (e.g., Event Detection aka Named Entity Recognition) has always remained a disjoint line of research from coreference resolution, despite the fact that the input of coreference resolution has always been the output of mention detection. Naturally, the performance of mention detection affects the eventual performance of coreference. Thus, it seems likely that merging these two into a single model could improve results, especially if one considers that (1) the confidence of two mentions co-referring could help estimate if the mentions are even valid mentions (e.g., “ran in the” should have low probability of corefering with any other mention, signifying that it is probably not a valid mention), and; (2) the confidence of a given text being a mention could help determine if two candidate mentions co-ref (e.g, “Barack Obama will” having a mention detection score of 0.5 and “Barack Obama” having a score of 0.95).

5.7 Related Work

Recently published [26] was the very first work which uses this idea. In short, the authors present the first end-to-end coreference model, whereby they consider all possible mention spans, and prune them based on boundaries learned from context during training. Notably, the model does not use third-party syntatic parse information. Instead, the only specific features used are: speaker, genre, span distance, mention width. Note, their work was for entity coreference and they used the CoNLL-2012 shared task, as opposed to our ECB+ corpus which does not have speaker or genre information.

5.8 Approach

A notable difference between our proposed work and the related work [26] is that we:

1. Want to consider *all* mention spans as candidates, and calculate coreference predictions with them, as opposed to pruning them before coreference.
2. Predict the mention *type* (e.g, action_occurrence, person, organization, etc) along with each span, in attempt to help coreference
3. Want to perform both entity and event resolution, as described in the previous section
4. Want to perform cross-document coreference, not just within-doc

The biggest challenge of these is arguably the 1^{st} item, as it is potentially prohibitely-expensive to compute all combination of candidate mention pairs, as this is $O(N^4)$. Current rough ideas for ways to approach this include:

- quick hashing techniques (e.g., MinHashing/LSH embeddings)
- heuristics (alpha-beta pruning) to stop exploring longer mention spans which have low scores from their constuite unigram, bi-gram candidate mentions
- try all mentions if possible (documents are short for the ECB+ corpus, so it might be possible, especially with parallel GPU jobs)

Specifically, I plan to evaluate our performance on (1) ECB+ corpus, as it has both entities and events labelled, and; (2) CoNLL-2012 for just entities, which will not leverage event coreference, as the corpus lacks this information, but it will give us a good comparison since this is the canonical corpus for entity coreference.

Bibliography

[1] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, Yiming Yang, James Allan Umass, Brian Archibald Cmu, Doug Beeferman Cmu, Adam Berger Cmu, Ralf Brown Cmu, Ira Carp Dragon, George Doddington Darpa, Alex Hauptmann Cmu, John Lafferty Cmu, Victor Lavrenko Umass, Xin Liu Cmu, Steve Lowe Dragon, Paul Van Mulbregt Dragon, Ron Papka Umass, Thomas Pierce Cmu, Jay Ponte Umass, and Mike Scudder Umass. Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998. Page 3.

[2] Amit Bagga and Breck Baldwin. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and Its Applications*, CorefApp '99, pages 1–8, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. Page 7.

[3] Cosmin Adrian Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1412–1422, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. Page 9.

[4] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 294–303, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. Page 8.

[5] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201, 1997. Page 5.

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 737–744. Morgan-Kaufmann, 1994. Page 12.

[7] Jason P. C. Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370, 2016. Page 5.

[8] Prafulla Kumar Choubey and Ruihong Huang. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *EMNLP*, 2017. Pages 8, 11, 17, and 19.

[9] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. *CoRR*, abs/1609.08667, 2016. Page 7.

[10] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations, 2016. cite arxiv:1606.01323Comment: Accepted for publication at the Association for Computational Linguistics (ACL), 2016. Pages 6 and 7.

[11] Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014. Page 8.

[12] Naomi Daniel, Dragomir Radev, and Timothy Allison. Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*, HLT-NAACL-DUC '03, pages 9–16, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. Page 3.

[13] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *EMNLP*, pages 1971–1982. ACL, 2013. Page 6.

[14] Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. A language-independent neural network for event detection. In *ACL*, 2016. Page 5.

[15] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. A convolutional encoder model for neural machine translation. In *ACL (1)*, pages 123–135. Association for Computational Linguistics, 2017. Page 12.

[16] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288, September 2002. Page 5.

[17] Ruslan Salakhutdinov Gregory Koch, Richard Zemel. Siamese neural networks for one-shot image recognition. In *ICML*, 2015. Page 12.

[18] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. Page 5.

[19] Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 385–393, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. Page 19.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. Page 5.

[21] Eduard H. Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. Events are not simple: Identity, non-identity, and quasi-identity. In *ACL*, 2013. Page 4.

[22] Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, ANARESOLUTION '97, pages 75–81, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. Pages 3 and 7.

[23] Richard Johansson and Pierre Nugues. Lth: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 227–230, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. Page 5.

[24] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751. ACL, 2014. Pages 12 and 13.

[25] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 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*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. Pages 9 and 19.

[26] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics, 2017. Pages 7 and 21.

[27] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 507–516, New York, New York, USA, 20–22 Jun 2016. PMLR. Page 14.

[28] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics. Page 5.

[29] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. Page 12.

[30] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. Page 5.

[31] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company. Page 5.

[32] Srini Narayanan and Sanda Harabagiu. Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. Page 3.

[33] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. Page 12.

[34] Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 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, 2014. Page 16.

[35] Vasin Punyakanok, Dan Roth, and Wen-tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist.*, 34(2):257–287, June 2008. Page 5.

[36] W.V. O. Quine. Events and reification. In *Action and Events: Perspectives on the philosophy of Donald Davidson*, pages 162–171, 1985. Page 4.

[37] Altaf Rahman and Vincent Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 814–824, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. Page 19.

[38] L. F. Rau. Extracting company names from text. In *Proc. of the Seventh Conference on Artificial Intelligence Applications CAIA-91 (Volume II: Visuals)*, pages 189–194, Miami Beach, FL, 1991. Page 5.

[39] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823. IEEE Computer Society, 2015. Page 14.

[40] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544, December 2001. Page 6.

[41] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 589–596, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. Page 5.

[42] Shyam Upadhyay, Nitish Gupta, Christos Christodoulopoulos, and Dan Roth. Revisiting the evaluation for cross document event coreference. In *COLING*, 2016. Page 4.

[43] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *HLT-NAACL*, pages 994–1004. The Association for Computational Linguistics, 2016. Pages 6 and 7.

[44] Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *ACL (1)*, pages 1416–1426. The Association for Computer Linguistics, 2015. Pages 5 and 7.

[45] Travis Wolfe, Mark Dredze, and Benjamin Van Durme. Predicate argument alignment using a global coherence model. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, 2015. Page 4.

[46] Bishan Yang, Claire Cardie, and Peter I. Frazier. A hierarchical distance-dependent bayesian model for event coreference resolution. *TACL*, 3:517–528, 2015. Pages 8, 10, 11, and 14.

[47] Xiang Yu and Ngoc Thang Vu. Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages. *CoRR*, abs/1705.10814, 2017. Page 12.