Towards Featureless Event Coreference Resolution via Conjoined Convolutional Networks

Chris Tanner and **Eugene Charniak**
Brown Linguistic Laboratory of Information Processing
Brown University
Providence, RI 02912
{christanner,ec}@cs.brown.edu

Abstract

Coreference resolution systems for entities and/or events almost always make use of many linguistic, parsing-based features. In contrast, (1) we introduce a new state-of-the-art event coreference resolution system which uses only lemmatization and its corresponding precomputed word-/char- embeddings, achieving 67.2 CoNLL F1 score on a common ECB+ test set, along with setting a new baseline of 8X.XX for the test set at large. (2) We exhaustively illustrate the performance of other commonly-used features. The crux of our system is that it first makes pairwise event-coreference predictions by using a Siamese Convolutional Neural Network (henceforth referred to as Conjoined Convolutional Neural Network or CCNN). Last, (3) we cluster the event mentions with a simple, but novel, neural approach which performs merges in an easy-first, cluster-holistic manner, allowing our system to be less susceptible to errors that are made exclusively from min-pairwise decisions.

1 Introduction

Coreference resolution is the task of trying to identify – within a single text or across multiple documents – which *mentions* refer to the same underlying discourse *entity* or *event*. Naturally, one may be solely interesting in determining if two given entities co-refer to the same object (e.g., a pairwise prediction of *she* and *Mary* co-referring); however, ultimately, coreference resolution is a clustering task, whereby we wish to group all like-mentions together. Successfully doing so can be useful for several other core NLP tasks that concern natural language understanding, such as information extraction (Humphreys et al., 1997), topic detection (Allan et al., 1998), text summarization (Daniel et al., 2003), knowledge base population (Mayfield and et al., 2009), question answering (Narayanan and Harabagiu, 2004), etc. Coreference Resolution has always been one of the fundamental tasks within NLP, and with the ever-increasing amount of textual, digital data that is generated and consumed in present-day, it remains both important and challenging.

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, the membership of which constitutes that every $m_i, m_j \in C_k$ is 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_k with a membership of one. Further, the number of distinct clusters is not known apriori but is bounded by the number of mentions and is part of the system's inference. Finding such a globally-optimal assignment is NP-Hard and thus computationally intractable. In attempt to avoid this, systems typically perform pairwise-mention predictions, then use those predictions to build up clusters. The specific modelling strategies for such approximately fall into two categories: (1) mention-ranking / mention-pairs; and (2) entity/event-level.

Mention-ranking models define a scoring a function $f(m_i, m_j)$ which operates on any 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). Although these models are by definition less expressive than entity/event-level models, their inference can be relatively simple and effective, allowing them to be fast and scalable. As a consequence, they have often been the approach used by many state-of-the-art systems (Soon et al., 2001; Durrett and Klein, 2013; Wiseman et al., 2016a).

Mention-pair models are defined almost identically, with the subtle difference being the target objective of the pairwise-candidates. That is, mention-ranking model aim to find the ideal m_i

antecedent for every m_j, whereas mention-pair models score all possible (m_i, m_j) pairs (Bengtson and Roth, 2008; Soon et al., 2001).

Entity/Event-level 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 – as opposed to the most local pairwise- elements that comprise the aforementioned models (Wiseman et al., 2016a; Clark and Manning, 2016b).

In this work, we use a novel and powerfully simple mention-ranking model that is designed solely to discriminate between pairs of input features: Siamese Convolutional Neural Networks, which, for political reasons, we will henceforth refer to as our newly-coined term, Conjoined Convolutional Neural Networks (or CCNN). Further, we aim to replace the main weakness of mention-ranking models with an approach resembling the main strength of entity/event-level models. Specifically, we aim to combine all linked mention pairs into a cluster via a simple neural, easy-first, clustering approach which factors in a small, but effective, notion of the entire cluster at large.

Additionally, a common theme of coreference research is that systems typically use a plethora of relatively-expensive parsing-based features, including dependency parse information, lemmatization, WordNet hypernyms/synonyms, FrameNet semantic roles, part-of-speech, etc. Although some research includes a listing of the learned feature weights of a system (corresponding to each feature's importance) (Yang et al., 2015), there has been a striking lack of work which takes the minimalistic approach and illustrates the effects of using few features. We aim to address this by starting with the widely-accepted strong baseline of *SameLemma* – two objects are co-referent if, and only if, they have the same lemmatization – and then evaluate the effectiveness of slowly adding other commonly used features.

Finally, in general, *entity* coreference resolution has received drastically more attention than *event* coreference. This lack of research could in part be due to events' often involving more complex nature: a single underlying event may be described via multiple lexicographically-differing mentions, yet different underlying events may also be represented by mentions that lexicographically look the same. This latter case is less common in entity coreference; other than pronouns, usually mentions' having the same text is a strong indication that the mentions are co-referent. In this paper, we are exclusively interested in event coreference.

In summary, we introduce a novel approach to event coreference resolution by performing mention-ranking with a Conjoined Convolutional Neural Network and unusually few features. We contribute a detailed performance analysis of other commonly used features. And last, we combine our predicted mention pairs into a cluster via a simple, neural approach which attempts to represent each cluster as a whole, yielding us with state-of-the-art results on the ECB+ corpus.

2 Related Work

Event coreference resolution has received significantly less attention than its entity-based counterpart. The seminal research on event-based 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. Most notable from this period were the works by Humphreys et al. (1997) and Bagga and Baldwin (1999), which applied event coreference to the tasks of information extraction, topic detection and tracking. The successor of MUC was the annual ACE program, which addressed more fine-grained events with the aforementioned challenging situations wherein mentions may have identical text shared by many distinct events.

In present day, Deep Learning is revolutionarily affecting NLP; however, there has been a few successful applications of deep learning to coreference resolution, almost all of which have been for entity-based system. We attribute this dearth to the fact that coreference resolution is inherently a clustering task, which tends to be a non-obvious modality for deep learning. We divide the work relevant to ours into two categories: (1) deep learning approaches; and (2) the systems which use the same ECB+ corpus (Cybulska and Vossen, 2014) as we do.

2.1 Deep Learning Approaches

To the best of our knowledge, there are only five other publications which apply deep learning to coreference resolution, four of which focus on entity coreference.

Sam Wiseman, et. al. built mention-ranking models (2015; 2016b) which are trained with a heuristic loss functions that assign different costs based on the types of errors made, and their latter work used mention-ranking predictions towards an entity-level model via LSTM hidden states (Hochreiter and Schmidhuber, 1997).

Separately, Clark and Manning (2016b; 2016a) also built both a mention-ranking 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 (2015) work.

2.2 Systems using ECB+ Corpus

For our research, we make use of the ECB+ corpus (Cybulska and Vossen, 2014), an extension of EventCorefBank (ECB) (Bejan and Harabagiu, 2010), which we further describe in Section 3.4. In short, this rich corpus provides annotations for both entities and events, yet most research chooses to focus 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. (2015) and Choubey's and Huang's Iterative-Unfolding approach (2017). Consequently both are highly relevant to our work.

2.2.1 HDDCRP Model

Yang, et. al's HDDCRP model (2015) uses a clever and inspiring mention-pair approach, whereby they first use logistic regression to train the set of parameters θ for the similarity function in Equation 1.

$$f_\theta(x_i, x_j) \propto \exp\{\theta^T \psi(m_i, m_j)\} \qquad (1)$$

Then, the crux of their system is that in a Chinese-restaurant-process fashion, they probabilistically assign links between mentions purely based on the scores provided by this similarity function. That is, the value emitted by $f(m_i, m_j)$ is directly correlated with the probability of (m_i, m_j) being chosen as a linked pair. However, identical to Bengtson's and Roth's work (Bengtson and Roth, 2008), the HDDCRP model then automatically 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 holding true for 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. After these initial clusters are formed for both within-doc (WD) and cross-document (CD) mentions, their system continues to perform Gibbs sampling until convergence, allowing mentions to freely shift between other clusters according to the similarity function. We aim to improve this shortcoming, as detailed in Section 5.

2.2.2 Neural Iterative-Unfolding Model

Most recently, Choubey and Huang (2017) introduced the first neural model that is exclusive for event coreference. Their system also fits into the mention-pair paradigm, whereby mentions are predicted by a feed-forward neural network. For within-doc predictions, their network features are primarily based on the cosine similarity and euclidean distance of input-pair embeddings. The cross-document model is identical, other than adding context features, too. This was an important finding, for they assert that when using the ECB+ corpus, within-doc coreference did not benefit from using mention context. That is, the mention words themselves were sufficient. Similar to the weakness of the HDDCRP model, they form clusters based on local mention-pair predictions, independent of mentions' relevance to the cluster at large.

3 System Overview

3.1 Mention Identification

Coreference systems are predicated upon having entity/event mentions identified. In fact, this identification process is the focus of a different line of research: entity recognition and event detection are the names given to identifying entities and events, respectively. This separation of tasks allows coreference systems to be evaluated precisely on their ability to link/cluster together appropriate mentions. Thus, it is common practice for coreference systems to either: (1) use gold mentions that are defined by the true annotations in the corpus, or (2) use an off-the-shelf entity recognition or semantic role labelling system. We do both. That is, the majority of our results are shown with having used gold mentions. Yet, it was critically important to us to ensure we developed a competitive system, so it was imperative to use the same mentions that were used in the two aforementioned system that focus on events of the ECB+ corpus. The HDDCRP model set the precedence by using an SRL system to predict mentions, then they pre-processed and filtered many of those, yielding their system with an imperfect but reasonable set of mentions that shares a moderate overlap with the gold mentions. Determining the exact mentions that were used by HDDCRP was one of the most challenging and time-consuming processes of our research.

Naturally, Choubey's, et. al. system also aimed to use their same mentions. After numerous exchanges with the author, it was clear that their set of mentions was similar and reasonable for research, but understandably not the same as that used by HDDCRP. Namely, they filtered out: (1) all predicted mentions which were not in the gold set (false positives), and (2) predicted mentions which were singletons (ones that did not cluster with a mention from another document).

We evaluate our systems having used the: (1) gold mentions; (2) HDDCRP-predicted mentions; (3) Choubey-predicted mentioned.

3.2 Reproducibility

As illustrated, reproducing coreference results can be naturally tedious, as it is challenging to ensure every token identified and parsed according to one system perfectly aligns and is represented correctly by another. Since these issues comprised a large amount of our research efforts, we aim to ameliorate the situation by providing our code online, which is easily runnable on any of the aforementioned sets of mentions and evaluations. Additionally, our code runs in just a few minutes on a single Titan X GPU.

3.3 Models

Our system is comprised of two neural models:

- Conjoined Convolutional Neural Network – used for making mention-pair predictions. (Section 4)
- Neural Clustering – uses the pairwise predictions to cluster mentions into events (Section 5)

3.4 Corpus

We exclusively make use of the ECB+ corpus (Cybulska and Vossen, 2014), which is the largest available dataset with annotations for event coreference. The corpus is comprised of 43 distinct *topics* – categories or news stories. Each of the 43 topics has 2 sub-topics which are similar in nature but distinctly different from each other. For example, Topic 1 contains 2 sub-topics, 1 of which about Lindsay Lohan checking into a rehab center in Malibu, California, and the other about Tara Reid checking into a rehab center in the same city. Each sub-topic contains roughly 8-15 short text documents which all concern the same given subtopic. Following the convention of the aforementioned researchers who use this corpus, we divide the corpus into the following splits: training set contains topics 1-20; dev set contains topics 21-23, and the test set contains topics 24-43. For those interested in this corpus, note that the actual structure of the corpus files happens to not include a topic directory named #15 or #17, so the listed divisions correspond to the sequential ordering of topics and size of each split, not the exact structure of directory names.

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

Table 1: Statistics of the ECB+ Corpus

Corpus statistics are listed in Table 1, where it is clear that the majority of gold mentions are one token in length (e.g, *announced*).

4 Conjoined Convolutional Neural Network

4.1 Motivation

As a recap, there has yet to exist a deep learning event-level model for event coreference resolution. Although it is tempting to explore this option due to the success of deep learning entity-based models, we were motivated for a few reasons to develop a model that fits the mention-pair paradigm: not only has the previous work on the ECB+ corpus shown strength from using mention-pair models, but analyzing the training set data illustrates that most golden co-referent clusters contain little variance in the lemma-representation of each mention (see Figure 1). Since there is such

Figure 1: A boat.

high homogeneity on an intra-cluster level, it suggests that entity-level representations might not offer much benefit. Naturally, one could argue that intra-cluster representation is not conclusive evidence; that is, the context of mentions, which could differ drastically for each co-referent mention, might be beneficial. However, as Choubey, et. al. (2017) concluded, context seems to offer not benefit at all for within-doc coreference on the ECB+ corpus. Thus, we are interested in developing a powerful pairwise-prediction model, such as a Conjoined Neural Network (Bromley et al., 1994).

4.2 Overview

Conjoined Neural Networks (or Siamese Networks, as they are known as) were first introduced by Bromly and LeCun (1994) towards a task whereby the goal was to accurately determine if two input items (hand signatures) were in fact of the same class or not. Specifically, a Conjoined Network can be defined as twin neural networks, each of which accepts distinct inputs, but they are eventually joined by a loss function over their highest-level features. The loss function computes a metric that represents the similarity between the two input pairs (e.g., euclidean distance, cosine similarity, hamming distance, etc). The two 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 into each network will be mapped accordingly, otherwise, they could be mapped to hidden representations that are disproportionately dissimilar from their input representations; and (2) forces the network to be symmetric. Namely, if we were to abstractly view the Conjoined Network as a function, then:

$$CCNN(f_i, f_j) \equiv CCNN(f_j, f_i)$$

This is critical, as the CCNN should yield the same similarity score independent of the ordering of its input pair.

Last, we posit that CCNN's have been shown to perform well in low-resource situation (Gregory Koch, 2015). This is ideal for our task, as it is highly likely that at test time we will encounter event mentions that are OOV. We desire our model to discriminately learn the relationships of input mentions, rather than exclusively relying on and memorizing the input values themselves.

As for the choice of Conjoined Network, Convolutional Neural Networks (CNNs) have recently proven to be highly useful for many tasks in NLP, including sentence classification (Kim, 2014), machine translation (Gehring et al., 2017), dependency parsing (Yu and Vu, 2017), etc. Likewise, we choose to use a CNN due to their power in learning sub-regions of features, and the relations thereof – rather than heavily relying manually-defined features.

4.3 Input Features

Since our CCNN needs each mention to be represented exclusively by its own input, we used none of the relational features that are common in other coreference systems (e.g., SameLemma, Jaccard Similarity of the mentions' context words, First common WordNet parent, # of Sentence in between Mentions, etc). We thoroughly tested the following input features and their listed variants:

- **Part-of-Speech:** 1-hot representation; LSTM-learned embeddings after maping the entire corpus to their POS tags
- **Lemmatization**: 300-length word embeddings for the lemma of each mention token, where the embedding came from running GLoVE (Pennington et al., 2014) either on our corpus, or using their provided pre-trained 6 billion and 840 billion token crawls.
- **Dependency Lemma:** we use the dependent parent and/or children of each mention token, and we represent it via the aforementioned lemma embeddings)
- **Character Embeddings:** we represent each mention token as a concatenation of its character embeddings (truncated or padded up to the first 20 characters), where we experimented with character embeddings being either (1) random 20-length embeddings or (2) pre-trained 20-length embeddings
- **Word Embeddings:** same as the embeddings listed for *lemma*, just we apply these embeddings for the word tokens themselves, not the lemma of each token.

Note, since mentions are of varying token length (see Table 1), we need a convention to standardize the vector-length (e.g., to 300 dimension). We experimented with summing across all token embeddings in place, averaging, and concatenated to a particular N-length size. For clarity, *averaging* for a given mention m's embedding is calculated via:

$m_{emb}[i] = \frac{\sum_{t \in T} t_{emb}[i]}{|T|}$,

where t is a token in the set of tokens T that comprise m

4.4 Architecture

We define the 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 for token t.

Naturally, we may want to convolve over the context of mention m, too, by including the N words before and after m. Thus, for a given window size of N, our entire matrix corresponding to mention m is of size $(2 * N + 1) \times d$, where d is the length of t_{emb}. Each row corresponds to a given token, and each column corresponds to a particular dimension in the vector space representation. A la Kim (2014), we zero-pad any tokens that would be beyond our window.

Let $\mathbf{M}$ represent the full matrix corresponding to mention m: $\mathbf{M} \in \mathbb{R}^{(2*N+1)\times d}$

and

$\mathbf{M}_{(i,j),(k:l)}$ represent the sub-matrix of M from (i, j) to (k, l).

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

$$c_i = f(\mathbf{w}^T \mathbf{M}_{(i:i+h-1),(j:j+w-1)} + b) \qquad (2)$$

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

We use several kernels (experimented with all powers of 2, from 2 - 256) and use ReLU as our activation function.

Dropout is then applied (experimented with values from 0 to 0.5).

Next, we repeat this processing by adding convolution and dropout again, then apply max-pooling to get a single $\hat{c} = max\{\mathbf{c}\} \in \mathbb{R}$.

Last, we apply dropout again, then merge all of our kernels to feed into a final ReLU.

4.5 Loss

4.6 Optimization

5 Neural Clustering

6 General Instructions

Manuscripts must be in two-column format. Exceptions to the two-column format include the title, authors' names and complete addresses, which must be centered at the top of the first page, and any full-width figures or tables (see the guidelines in Subsection 6.6). **Type single-spaced.** Start all pages directly under the top margin. See the guidelines later regarding formatting the first page. The manuscript should be printed single-sided and its length should not exceed the maximum page limit described in Section **??**. Pages are numbered for initial submission. However, **do not number the pages in the camera-ready version**.

By uncommenting `\aclfinalcopy` at the top of this document, it will compile to produce an example of the camera-ready formatting; by leaving it commented out, the document will be anonymized for initial submission. When you first create your submission on softconf, please fill in your submitted paper ID where `***` appears in the `\def\aclpaperid{***}` definition at the top.

The review process is double-blind, so do not include any author information (names, addresses) when submitting a paper for review. However, you should maintain space for names and addresses so that they will fit in the final (accepted) version. The NAACL-HLT 2018 LaTeX style will create a titlebox space of 2.5in for you when `\aclfinalcopy` is commented out.

The author list for submissions should include all (and only) individuals who made substantial contributions to the work presented. Each author listed on a submission to NAACL-HLT 2018 will be notified of submissions, revisions and the final decision. No authors may be added to or removed from submissions to NAACL-HLT 2018 after the submission deadline.

6.1 The Ruler

The NAACL-HLT 2018 style defines a printed ruler which should be presented in the version submitted for review. The ruler is provided in order that reviewers may comment on particular lines in the paper without circumlocution. If you are preparing a document without the provided style files, please arrange for an equivalent ruler to appear on the final output pages. The presence or absence of the ruler should not change the appearance of any other content on the page. The camera ready copy should not contain a ruler. (LaTeX users may uncomment the `\aclfinalcopy` command in the document preamble.)

Reviewers: note that the ruler measurements do not align well with lines in the paper – this turns out to be very difficult to do well when the paper contains many figures and equations, and, when done, looks ugly. In most cases one would expect that the approximate location will be adequate, although you can also use fractional references (*e.g.*, the first paragraph on this page ends at mark 108.5).

6.2 Electronically-available resources

NAACL-HLT provides this description in LaTeX2e (`naaclhlt2018.tex`) and PDF format (`naaclhlt2018.pdf`), along with the LaTeX2e style file used to format it (`naaclhlt2018.sty`) and an ACL bibliography style (`acl_natbib.bst`) and example bibliography (`naaclhlt2018.bib`). These files are all available at `http://naacl2018.org/downloads/naaclhlt2018-latex.zip`. We strongly recommend the use of these style files, which have been appropriately tailored for the NAACL-HLT 2018 proceedings.

6.3 Format of Electronic Manuscript

For the production of the electronic manuscript you must use Adobe's Portable Document Format (PDF). PDF files are usually produced from LaTeX using the *pdflatex* command. If your version of LaTeX produces Postscript files, you can convert these into PDF using *ps2pdf* or *dvipdf*. On Windows, you can also use Adobe Distiller to generate PDF.

Please make sure that your PDF file includes all the necessary fonts (especially tree diagrams, symbols, and fonts with Asian characters). When you print or create the PDF file, there is usually an option in your printer setup to include none, all or just non-standard fonts. Please make sure that you select the option of including ALL the fonts. **Before sending it, test your PDF by printing it from a computer different from the one where it was created.** Moreover, some word processors may generate very large PDF files, where each page is rendered as an image. Such images may reproduce poorly. In this case, try alternative ways to obtain the PDF. One way on some systems is to install a driver for a postscript printer, send your document to the printer specifying "Output to a file", then convert the file to PDF.

It is of utmost importance to specify the **A4 format** (21 cm x 29.7 cm) when formatting the paper. When working with `dvips`, for instance, one should specify `-t a4`. Or using the command `\special{papersize=210mm,297mm}` in the latex preamble (directly below the `\usepackage` commands). Then using `dvipdf` and/or `pdflatex` which would make it easier for some.

Print-outs of the PDF file on A4 paper should be identical to the hardcopy version. If you cannot meet the above requirements about the production of your electronic submission, please contact the publication chairs as soon as possible.

6.4 Layout

Format manuscripts two columns to a page, in the manner these instructions are formatted. The exact dimensions for a page on A4 paper are:

- Left and right margins: 2.5 cm
- Top margin: 2.5 cm
- Bottom margin: 2.5 cm
- Column width: 7.7 cm
- Column height: 24.7 cm
- Gap between columns: 0.6 cm

Papers should not be submitted on any other paper size. If you cannot meet the above requirements about the production of your electronic submission, please contact the publication chairs above as soon as possible.

6.5 Fonts

For reasons of uniformity, Adobe's **Times Roman** font should be used. In LaTeX2e this is accomplished by putting

Type of Text	Font Size	Style
paper title	15 pt	bold
author names	12 pt	bold
author affiliation	12 pt	
the word “Abstract”	12 pt	bold
section titles	12 pt	bold
document text	11 pt	
captions	10 pt	
abstract text	10 pt	
bibliography	10 pt	
footnotes	9 pt	

Table 2: Font guide.

```
\usepackage{times}
\usepackage{latexsym}
```

in the preamble. If Times Roman is unavailable, use **Computer Modern Roman** (LaTeX2e's default). Note that the latter is about 10% less dense than Adobe's Times Roman font.

6.6 The First Page

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Do not include the paper ID number assigned during the submission process. Use the two-column format only when you begin the abstract.

Title: Place the title centered at the top of the first page, in a 15-point bold font. (For a complete guide to font sizes and styles, see Table 2) Long titles should be typed on two lines without a blank line intervening. Approximately, put the title at 2.5 cm from the top of the page, followed by a blank line, then the author's names(s), and the affiliation on the following line. Do not use only initials for given names (middle initials are allowed). Do not format surnames in all capitals (*e.g.*, use “Mitchell” not “MITCHELL”). Do not format title and section headings in all capitals as well except for proper names (such as “BLEU”) that are conventionally in all capitals. The affiliation should contain the author's complete address, and if possible, an electronic mail address. Start the body of the first page 7.5 cm from the top of the page.

The title, author names and addresses should be completely identical to those entered to the electronical paper submission website in order to maintain the consistency of author information among all publications of the conference. If they

Command	Output
`{\"a}`	ä
`{\^e}`	ê
`{\`i}`	ì
`{\.I}`	İ
`{\o}`	ø
`{\'u}`	ú
`{\aa}`	å

Command	Output
`{\c c}`	ç
`{\u g}`	ğ
`{\l}`	ł
`{\~n}`	ñ
`{\H o}`	ő
`{\v r}`	ř
`{\ss}`	ß

Table 3: Example commands for accented characters, to be used in, *e.g.*, BIBTeX names.

are different, the publication chairs may resolve the difference without consulting with you; so it is in your own interest to double-check that the information is consistent.

Abstract: Type the abstract at the beginning of the first column. The width of the abstract text should be smaller than the width of the columns for the text in the body of the paper by about 0.6 cm on each side. Center the word **Abstract** in a 12 point bold font above the body of the abstract. The abstract should be a concise summary of the general thesis and conclusions of the paper. It should be no longer than 200 words. The abstract text should be in 10 point font.

Text: Begin typing the main body of the text immediately after the abstract, observing the two-column format as shown in

the present document. Do not include page numbers.

Indent: Indent when starting a new paragraph, about 0.4 cm. Use 11 points for text and subsection headings, 12 points for section headings and 15 points for the title.

6.7 Sections

Headings: Type and label section and subsection headings in the style shown on the present document. Use numbered sections (Arabic numerals) in order to facilitate cross references. Number subsections with the section number and the subsection number separated by a dot, in Arabic numerals. Do not number subsubsections.

Citations: Citations within the text appear in parentheses as (Gusfield, 1997) or, if the author's name appears in the text itself, as Gusfield (1997). Using the provided LaTeX style, the former is accomplished using `\cite` and the latter with `\shortcite` or `\newcite`. Collapse multiple citations as in (Gusfield, 1997; Aho and Ullman, 1972); this is accomplished with the provided style using commas within the `\cite` command, *e.g.*, `\cite{Gusfield:97,Aho:72}`. Ap-

output	natbib	previous ACL style files
(Gusfield, 1997)	`\citep`	`\cite`
Gusfield (1997)	`\citet`	`\newcite`
(1997)	`\citeyearpar`	`\shortcite`

Table 4: Citation commands supported by the style file. The citation style is based on the natbib package and supports all natbib citation commands. It also supports commands defined in previous ACL style files for compatibility.

pend lowercase letters to the year in cases of ambiguities. Treat double authors as in (Aho and Ullman, 1972), but write as in (Chandra et al., 1981) when more than two authors are involved. Collapse multiple citations as in (Gusfield, 1997; Aho and Ullman, 1972). Also refrain from using full citations as sentence constituents.

We suggest that instead of

> "(Gusfield, 1997) showed that ..."

you use

> "Gusfield (1997) showed that ..."

If you are using the provided LaTeX and BibTeX style files, you can use the command `\citet` (cite in text) to get "author (year)" citations.

If the BibTeX file contains DOI fields, the paper title in the references section will appear as a hyperlink to the DOI, using the hyperref LaTeX package. To disable the hyperref package, load the style file with the `nohyperref` option:
`\usepackage[nohyperref]{naaclhlt2018}`

Digital Object Identifiers: As part of our work to make ACL materials more widely used and cited outside of our discipline, ACL has registered as a CrossRef member, as a registrant of Digital Object Identifiers (DOIs), the standard for registering permanent URNs for referencing scholarly materials. As of 2017, we are requiring all camera-ready references to contain the appropriate DOIs (or as a second resort, the hyperlinked ACL Anthology Identifier) to all cited works. Thus, please ensure that you use BibTeX records that contain DOI or URLs for any of the ACL materials that you reference. Appropriate records should be found for most materials in the current ACL Anthology at `http://aclanthology.info/`.

As examples, we cite (Goodman et al., 2016) to show you how papers with a DOI will appear in the bibliography. We cite (Harper, 2014) to show how papers without a DOI but with an ACL Anthology Identifier will appear in the bibliography.

As reviewing will be double-blind, the submitted version of the papers should not include the authors' names and affiliations. Furthermore, self-references that reveal the author's identity, *e.g.*,

> "We previously showed (Gusfield, 1997) ..."

should be avoided. Instead, use citations such as

> "Gusfield (1997) previously showed ... "

Any preliminary non-archival versions of submitted papers should be listed in the submission form but not in the review version of the paper. NAACL-HLT 2018 reviewers are generally aware that authors may present preliminary versions of their work in other venues, but will not be provided the list of previous presentations from the submission form.

Please do not use anonymous citations and do not include when submitting your papers. Papers that do not conform to these requirements may be rejected without review.

References: Gather the full set of references together under the heading **References**; place the section before any Appendices, unless they contain references. Arrange the references alphabetically by first author, rather than by order of occurrence in the text. Provide as complete a citation as possible, using a consistent format, such as the one for *Computational Linguistics* or the one in the *Publication Manual of the American Psychological Association* (American Psychological Association, 1983). Use of full names for authors rather than initials is preferred. A list of abbreviations for common computer science journals can be found in the ACM *Computing Reviews* (for Computing Machinery, 1983).

The LaTeX and BibTeX style files provided roughly fit the American Psychological Association format, allowing regular citations, short citations and multiple citations as described above.

Submissions should accurately reference prior and related work, including code and data. If a piece of prior work appeared in multiple venues, the version that appeared in a refereed, archival venue should be referenced. If multiple versions of a piece of prior work exist, the one used by the authors should be referenced. Authors should not rely on automated citation indices to provide accurate references for prior and related work.

Appendices: Appendices, if any, directly follow the text and the references (but see above). Letter them in sequence and provide an informative title: **Appendix A. Title of Appendix**.

6.8 Footnotes

Footnotes: Put footnotes at the bottom of the page and use 9 point font. They may be numbered or referred to by asterisks or other symbols.[1] Footnotes should be separated from the text by a line.[2]

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

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. 1998. Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. pages 194–218.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and Its Applications*. Association for Computational Linguistics, Stroudsburg, PA, USA, CorefApp '99, pages 1–8. http://dl.acm.org/citation.cfm?id=1608810.1608812.

Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 1412–1422. http://dl.acm.org/citation.cfm?id=1858681.1858824.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 294–303. http://dl.acm.org/citation.cfm?id=1613715.1613756.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. 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*, Morgan-Kaufmann, pages 737–744. http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-tim pdf.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery* 28(1):114–133. https://doi.org/10.1145/322234.322243.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *EMNLP*. http://www.aclweb.org/anthology/D17-1225.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. *CoRR* abs/1609.08667. http://cs.stanford.edu/people/kevclark/resources/clark-manning-emnlp2016-deep.pdf.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. Cite arxiv:1606.01323Comment: Accepted for publication at the Association for Computational Linguistics (ACL), 2016. http://arxiv.org/abs/1606.01323.

Agata Cybulska and Piek Vossen. 2014. 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). http://www.aclweb.org/anthology/L14-1646.

[1] This is how a footnote should appear.

[2] Note the line separating the footnotes from the text.

Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT-NAACL-DUC '03, pages 9–16. https://doi.org/10.3115/1119467.1119469.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*. ACL, pages 1971–1982. http://nlp.cs.berkeley.edu/pubs/Durrett-Klein_2013_Coreference_paper.pdf.

Association for Computing Machinery. 1983. *Computing Reviews* 24(11):503–512.

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

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1–11. https://doi.org/10.18653/v1/P16-1001.

Richard Zemel Ruslan Salakhutdinov Gregory Koch. 2015. Siamese neural networks for one-shot image recognition. http://www.cs.toronto.edu/~rsalakhu/papers/oneshot1.pdf.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, page 1. http://aclweb.org/anthology/C14-1001.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*. Association for Computational Linguistics, Stroudsburg, PA, USA, ANARESOLUTION '97, pages 75–81. http://dl.acm.org/citation.cfm?id=1598819.1598830.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. ACL, pages 1746–1751. http://aclweb.org/anthology/D/D14/D14-1181.pdf.

James Mayfield and et al. 2009. Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. In *Proceedings of the AAAI 2009 Spring Symposium on Learning by Reading and Learning to Read*. AAAI Press. http://ebiquity.umbc.edu/_file_directory_/papers/442.pdf.

Srini Narayanan and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '04. https://doi.org/10.3115/1220355.1220455.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27(4):521–544. http://dl.acm.org/citation.cfm?id=972597.972602.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016a. Learning global features for coreference resolution. *CoRR* abs/1604.03035. http://nlp.seas.harvard.edu/papers/corefmain.pdf.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016b. Learning global features for coreference resolution. In *HLT-NAACL*. The Association for Computational Linguistics, pages 994–1004. http://aclweb.org/anthology/N/N16/N16-1114.pdf.

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *ACL (1)*. The Association for Computer Linguistics, pages 1416–1426. http://aclweb.org/anthology/P/P15/P15-1137.pdf.

Bishan Yang, Claire Cardie, and Peter I. Frazier. 2015. A hierarchical distance-dependent bayesian model for event coreference resolution. *TACL* 3:517–528. http://dblp.uni-trier.de/db/journals/tacl/tacl3.html#YangCF15.

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

A Supplemental Material

Submissions may include resources (software and/or data) used in in the work and described in the paper. Papers that are submitted with accompanying software and/or data may receive additional credit toward the overall evaluation score, and the potential impact of the software and data will be taken into account when making the acceptance/rejection decisions. Any accompanying software and/or data should include licenses and documentation of research review as appropriate.

NAACL-HLT 2018 also encourages the submission of supplementary material to report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. **Submissions that misuse the supplementary material may be rejected without review.** Essentially, supplementary material may include explanations or details of proofs or derivations that do not fit into the paper, lists of features or feature templates, sample inputs and outputs for a system, pseudo-code or source code, and data. (Source code and data should be separate uploads, rather than part of the paper).

The paper should not rely on the supplementary material: while the paper may refer to and cite the supplementary material and the supplementary material will be available to the reviewers, they will not be asked to review the supplementary material.

Appendices (*i.e.* supplementary material in the form of proofs, tables, or pseudo-code) should come after the references, as shown here. Use `\appendix` before any appendix section to switch the section numbering over to letters.

B Multiple Appendices

. . . can be gotten by using more than one section. We hope you won't need that.