# Coreference Resolution

## Working Title, First Draft: Content, not Wording

Patrick Kahardipraja, Olena Vyshnevska

## 1 Introduction

Coreference Resolution (CR) is a vital part of Natural Language Understanding and Natural Language Processing (NLP). It's applications include information extraction, question answering, summarization, machine translation, dialog systems etc. The task consists of mention detection and mention clustering. Some systems only focus on one part, while others propose end-to-end solutions.

The notion of mention – a span of text referring to some entity – is central to the task of CR. Mentions can be one of the three kinds: pronouns, named entity, and noun phrases. When several mentions *corefer* they refer to the same entity.

Since the seventies automated solutions for coreference resolutions have been researched (Woods, Kaplan, and Nash-webber 1972; Winograd 1972; Hobbs 1978). Some of the challenges central to the field include semantic and syntactic agreement between mentions; encoding non-local dependencies; paraphrasing of repetitions.

Since the introduction of pre-trained BERT model (Devlin et al. 2019) there has been a lot of research showing how the model outperforms previous state-of-the-art task-specific NLP models. cite some papers here Joshi et al. (2019) has introduced BERT models into coreference resolution tasks. However, the researchers have also encoded other information on top of BERT embeddings.

Our first research question is based on the recent research which shows that the BERT model itself encodes a significant amount of semantic and syntactic information. Therefore, we decided to test how well the span representations using BERT embeddings from the model by Joshi et al. (2019) finetuned on Ontonotes dataset (Pradhan et al. 2012) can encode coreference resolution information without any layers of additional information about a given sequence.

As a baseline, we used original pre-trained BERT embeddings. To encode a mention span, we used a convolutional layer and a self-attention layer.

Our second research question, which naturally follows from the first one, whether the span representations encoded with a pre-trained BERT model simply modelling local coreference? Will it be able to encode non-local coreference dependencies?

We found that the span represenations from within the coreference model by Joshi et al. (2019) reach *insert some numbers*. The baseline model is at *insert numbers*. This findings suggests that the pre-trained BERT model is a powerful tool for encoding information relevant to coreference resolution.

For our second research question (non-local VS local dependencies) we found that ... what we found for local non-locat

Our code is publicly available on GitHub. [1].

[1] https://github.com/alyonavyshnevska/bert_for_coreference_resolution

# 2 Related Work

## 2.1 Approaches to Coreference Resolution

Architectures for coreference resolution models are typically categorized as 1) mention-pair classifiers (Ng and Cardie 2002; Bengtson and Roth 2008), 2) mention-ranking classifiers (Durrett and Klein 2013; Wiseman, Rush, S. Shieber, et al. 2015; Clark and Manning 2016), 3) entity-level models (Haghighi and Klein 2010; Wiseman, Rush, and S. M. Shieber 2016), or 4) latent-tree models (Fernandes, Santos, and Milidiu 2012; Martschat and Strube 2015). Earlier solutions have been feature-based, while in the recent years neural classifiers have been particularly successful.

Nowadays a widely-adopted approach to coreference resolution are end-to-end models that perform mention detection, anaphoricity, and coreference jointly (Daniel Jurafsky 2019). Beforehand, the rule-based systems were in use. Notable mentions are work by Hobbs (1978), Lappin and Leass (1994). Hobbs (1978) invented a tree-search algorithm for identifying reference with robust results, which started a long series of syntax-based methods. Lappin and Leass (1994) combined syntactic and other features by assigning weights to those features and summing these up to score candidate mentions. Kennedy and Boguraev (1996) optimised the approach to avoid full syntactic parsing.

Yang et al. (2003) and Iida et al. (2003) helped establish mention-ranking approaches as influential solutions in the early 2000's. Ng (2005) pioneered the rise of end-to-end solutions. While rule-based systems have witnessed a short-lived revival in 2010s (Zhou, Ticrea, and Hovy 2004; Haghighi and Klein 2009), their struggles with semantic understanding for the models led to their eventual demise. The rise of neural architectures that dominated the NLP in the recent decade has inevitably established itself in coreference.

## 2.2 Coreference Arc Prediction Task

Recently, there has been interesting work done to simplify state-of-the-art task-specific NLP models. Liu et al. (2019) show evidence that frozen contextual represenations of word sequences fed into linear models can show similar levels of performance as state-of-the-art task-specific models on many NLP tasks. The authors use probing models, also known as auxiliary or diagnostic classifiers (Shi, Padhi, and Knight 2016; Kádár, Chrupała, and Alishahi 2017) to analyse the linguistic information within contextual word representations. To test how the probing model performs in comparison to ELMo, the authors tested the models on a coreference arc prediction task, where the model predicts whether two mentions corefer. The Ontonotes dataset was used. To train the probing model the authors generate negative examples, where mentions that do not corefer are fed into the probing model alongside the mentions that do corefer. In our work we will also build our experiments similar to coreference arc prediction tasks suggested by the authors.

Tenney et al. (2019) introduced edge probing task design. For coreference resolution, the models had to determining whether two spans of tokens refer to the same entity. In order to investigate how good the contextual word representation models can model long-range dependencies, the authors used a fixed-width convolutional layer on top of the word representations to build spans of word pieces. The authors concluded that using the CNN layers on top of BERT-large pretarined model has performed particularly well on difficult semantic tasks, such as coreference resolution. Hence, in out work we will also follow the approach of the authors to construct our baseline.

## 3 Overview of Span-Ranking Architecture

The first coreference resolution model that is trained in an end-to-end manner without relying on any syntactic parser or hand-engineered mention detector is introduced by (Lee, L. He, Lewis, et al. 2017). To construct correct coreference clusters, the neural model learn to jointly model mention detection and coreference prediction using span-ranking approach. With span-ranking approach, the model is able to consider all spans in a document up to a defined maximum length as a candidate mention and select the corresponding antecedent for each mention based on the antecedent distribution. We will further refer to the model as *e2e-coref* in the paper.

The span-ranking architecture estimates the joint probability distribution of mention-spans that belong to the same cluster given the document, by calculating product of multinomials for each span:

$$P(y_1, ..., y_n|D) = \prod_{i=1}^{N} P(y_i|D) \tag{1}$$

$$= \prod_{i=1}^{N} \frac{\text{exp}(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \text{exp}(s(i, y'))} \tag{2}$$

where $s(i, j)$ is a pairwise score for a coreference link between span $i$ and span $j$ in document $D$. The coreference score is composed of three factors: (1) $s_m(i)$, mention score of $i$ (i.e. how possible is the span $i$ to be a mention), (2) $s_m(j)$, mention score of $j$ and (3) $s_a(i, j)$, joint score of $i$ and $j$ (i.e. assuming that $i$ and $j$ are both mentions, how possible that they both refer to the same entity), formulated as the following:

$$s(i, j) = \begin{cases} s_m(i) + s_m(j) + s_a(i, j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases} \tag{3}$$

where $\epsilon$ is a dummy antecedent to represent that the span is not a mention or the case where it is a mention that coreferent with none of the previous mention span.

In this model, a bidirectional LSTM (Hochreiter and Schmidhuber 1997) is used to compute vector representations $\boldsymbol{g}_i$ of each possible span $i$ from the word and character embeddings in the span. The word embeddings itself are a fixed concatenation of GloVe (Pennington, Socher, and Manning 2014) and Turian embeddings (Turian, Ratinov, and Bengio 2010). The character embeddings are learned using CNN, with three various window sizes. It also provide morphological information and allows the model to deal with rare words. The span representation $\boldsymbol{g}_i$, which we will describe in detail in the following section, is then used to compute $s_m(i)$ and $s_a(i, j)$ via feed-forward neural networks as follows:

$$s_m(i) = \boldsymbol{w}_m \cdot \text{FFNN}_m(\boldsymbol{g}_i) \tag{4}$$

$$s_a(i, j) = \boldsymbol{w}_a \cdot \text{FFNN}_a([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \odot \boldsymbol{g}_j, \phi(i, j)]) \tag{5}$$

where $\odot$ denotes the Hadamard product between $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$, and $\phi(i, j)$ is a feature vector that encodes speaker and genre information from the metadata and the distance between a pair of span. The final coreference score is computed through a two-stage beam search as the model complexity is quartic in terms of the document length. In the first phase, a beam of up to $\lambda T$ candidate mention spans are considered based on the highest mention scores $s_m(i)$, where $\lambda$ is a hyperparameter and $T$ is the document length. Afterwards, only the filtered mentions are used to compute the joint score $s_a(i, j)$ during both training and inference, with the limitation that only up to $K$ candidate antecedents that are considered for each mention.

While the first order model proposed by (Lee, L. He, Lewis, et al. 2017) works quite well for coreference resolution, scoring only pairs of entity mentions means that previous coreference decisions are not taken into account when computing coreference decisions that occur afterwards. This drawback makes the model more susceptible to predicting clusters that are locally consistent but globally inconsistent. In an attempt to improve the weakness of this approach, (Lee, L. He, and Zettlemoyer 2018) proposed a model that captures higher-order interactions between mention spans in predicted coreference clusters. The model utilizes span-ranking architecture to refine existing span representations iteratively with the antecedent distribution as an attention mechanism. We will further refer to the model as *c2f-coref* in the paper.

In order to refine the span representations, the higher-order model estimates the antecedent distributions $P_n(y_i)$, which is obtained using the softmax function:

$$P_n(y_i) = \frac{\exp(s(\boldsymbol{g}_i^n, \boldsymbol{g}_{y_i}^n))}{\sum_{y \in \mathcal{Y}(i)} \exp(s(\boldsymbol{g}_i^n, \boldsymbol{g}_y^n))} \tag{6}$$

where $s$ is the scoring function from (Lee, L. He, Lewis, et al. 2017), $\boldsymbol{g}_i^n$ is the representation for span $i$ at iteration $n$. The refined span representations also allow the model to iteratively refine $P_n(y_i)$. Afterwards, the expectation of each span $i$ are computed by using the current antecedent distribution as the attention weight:

$$\boldsymbol{a}_i^n = \sum_{y_i \in \mathcal{Y}(i)} P_n(y_i) \cdot \boldsymbol{g}_{y_i}^n \tag{7}$$

where $\boldsymbol{a}_i^n$ can be viewed as the expected antecedent representation. The span representation $\boldsymbol{g}_i^n$ is updated with the expected antecedent representation $\boldsymbol{a}_i^n$ using coupled forget and input gates formulated as the following:

$$\boldsymbol{f}_i^n = \sigma(\mathbf{W}_f[\boldsymbol{g}_i^n, \boldsymbol{a}_i^n]) \tag{8}$$

$$\boldsymbol{g}_i^{n+1} = \boldsymbol{f}_i^n \odot \boldsymbol{g}_i^n + (\mathbf{1} - \boldsymbol{f}_i^n) \odot \boldsymbol{a}_i^n \tag{9}$$

where $\boldsymbol{f}_i^n \in [0, 1]$ is a gate vector that controls whether to retain the information from the current span representation or to update the span representation with the expected antecedent. Altogether, this mechanism allows the model to compute the final antecedent distribution conditioned on up to $n$ other mention spans.

To alleviate the additional computational cost of higher-order inference, (Lee, L. He, and Zettlemoyer 2018) also proposed a coarse-to-fine beam search, which does not establish an a priori maximum coreference distance. In order to prune potential mentions aggresively, the coreference score is reformulated as the following:

$$s_c(i, j) = \boldsymbol{g}_i^\top \mathbf{W}_c \boldsymbol{g}_j \tag{10}$$

$$s(i, j) = \begin{cases} s_m(i) + s_m(j) + s_a(i, j) + s_c(i, j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases} \tag{11}$$

where $s_c(i, j)$ a bilinear scoring function to compute a rough sketch of possible antecedents. To obtain the coreference score, the model executes a three-stage beam search. At the first stage, a beam of up to $M$ candidate mention spans are considered based on the mention score $s_m(i)$. Afterwards, approximation of pairwise coreference score is computed based on $s_m(i) + s_m(j) + s_c(i, j)$ to keep top $K$ candidate antecedents for each remaining mention spans. In the last phase, the final coreference score $s(i, j)$ is computed based on the surviving pairs of mention spans.

Recently, BERT (Devlin et al. 2019) has achieved state-of-the-art results on a wide array of NLP tasks, such as question answering and natural language inference, without any substantial task-specific architecture modifications. (Joshi et al. 2019) proposed to replace the bidirectional LSTM encoder in *c2f-coref* with BERT transformers and fine-tune it on coreference resolution task. Although BERT improves the state-of-the-art results in other tasks significantly, coreference resolution still proves to be a challenging task, as BERT encoder offers only a marginal performance increase. Furthermore, the model still struggles in modeling pronouns and resolving cases where mention paraphrasing is required. We refer the model by (Joshi et al. 2019) as *BERT-coref*.

# 4 Probing Mention-span Representations

## 4.1 Span Representation

Span representation plays an important role in span-ranking model (Lee, L. He, Lewis, et al. 2017), since it is used to compute distribution over candidate antecedent spans. In order to be able to predict coreference relations accurately, span representation should also capture the information of the span's internal structure and its surrounding context. In our experiments, we use the span representation which is first proposed in (Lee, L. He, Lewis, et al. 2017), but with BERT (Devlin et al. 2019) instead of a LSTM-based encoder to encode lexical information of the span and its surrounding, following (Joshi et al. 2019). The span representation is a vector embeddings which consists of context-dependent boundary representations with attentional representation of the head words over the span. The boundary representations are composed of first and last word-pieces of the span itself. The head words are automatically learned using additive attention (Bahdanau, Cho, and Bengio 2015) over each word-pieces in the span:

$$\alpha_t = \boldsymbol{w}_\alpha \cdot \text{FFNN}_\alpha(\boldsymbol{x}_t^*) \tag{12}$$

$$a_{i,t} = \frac{exp(\alpha_t)}{\sum\limits_{k=start(i)}^{end(i)} exp(\alpha_k)} \tag{13}$$

$$\hat{\boldsymbol{x}}_i = \sum_{t=start(i)}^{end(i)} a_{i,t} \cdot \boldsymbol{x}_t \tag{14}$$

where $\hat{\boldsymbol{x}}_i$ is a weighted vector representation of word-pieces for span $i$. This representation is augmented by a $\mathbb{R}^d$ feature vector which encodes the size of span $i$ with $d = 20$. The final representation $\boldsymbol{g}_i$ for span $i$ is formulated as the following:

$$\boldsymbol{g}_i = [\boldsymbol{x}^*_{start(i)}, \boldsymbol{x}^*_{end(i)}, \hat{\boldsymbol{x}}_i, \phi_i] \tag{15}$$

where $\boldsymbol{x}^*_{start(i)}$ and $\boldsymbol{x}^*_{end(i)}$ are first and last word-pieces of the span, and $\phi_i$ is the span width embeddings.

## 4.2 Coreference Arc Prediction

We focus on coreference arc prediction task, which is a part of probing tasks suite for contextual word embeddings (Liu et al. 2019; Tenney et al. 2019). In this task, a probing model is trained to determine whether if two mentions refer to the same entity. As datasets for coreference resolution usually only contains annotations for corefering mentions, we produce negative samples following the approach by (Liu et al. 2019). For every pair of mentions $(w_i, w_j)$, where they belong to the same coreference cluster and $w_i$ is an antecedent of $w_j$, we generate a negative example $(w_{random}, w_j)$ where $w_{random}$ belongs to a different coreference cluster. This method ensures that the ratio between positive and negative examples is balanced. We also follow the approach of (Tenney et al. 2019) by using spans of word-pieces for mentions, as (Liu et al. 2019)'s approach is limited on single-token mentions and therefore unable to fully exploit the information in the mention-span.

### 4.3 Probing Model

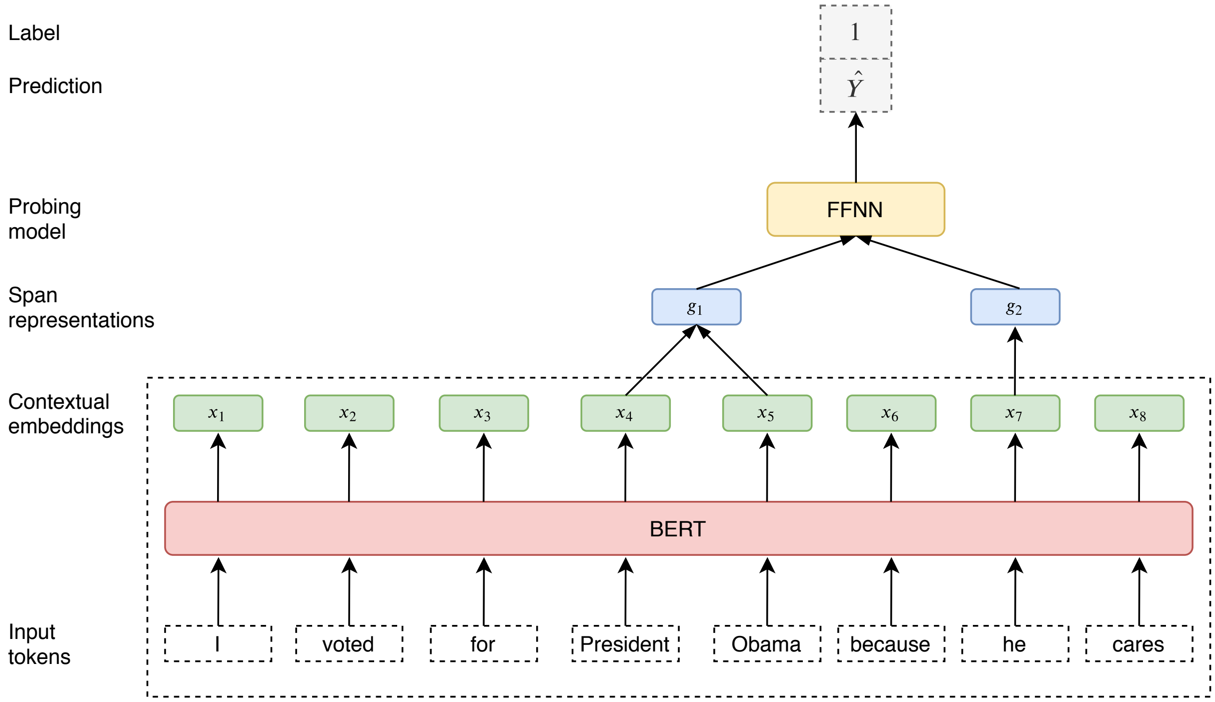


Figure 1: Probing architecture for span representation. The feed-forward neural network is trained to extract information from the span representations $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, while all the parameters inside the dashed line are frozen. The example depicts a mention-pair, where $\boldsymbol{g}_1$ corresponds to span representation of "President Obama", while $\boldsymbol{g}_2$ corresponds to "he". We predict $\hat{Y}$ as positive for this example.

Our probing model is a simple feed-forward neural network, which is designed with a limited capacity to focus on what information that can be extracted from the span representations. As the input to our probing model, we take the span representation for pair of mention-spans $\boldsymbol{g}_1 = [\boldsymbol{x}^*_{start(1)}, \boldsymbol{x}^*_{end(1)}, \hat{\boldsymbol{x}}_1, \phi_1]$ and $\boldsymbol{g}_2 = [\boldsymbol{x}^*_{start(2)}, \boldsymbol{x}^*_{end(2)}, \hat{\boldsymbol{x}}_2, \phi_2]$, where both $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are concatenated together and passed to the FFNN. The FFNN consists of a single hidden layer followed by a sigmoid output layer. The model is trained to minimize the binary cross-entropy with respect to the gold label $Y \in \{0, 1\}$. The probing architecture is shown in Figure 1.

We explore mention-span representations obtained from BERT. BERT (Devlin et al. 2019) is a language representations model which uses deep Transformer architecture (Vaswani et al. 2017), trained jointly with masked language model and next sentence prediction objective on BooksCorpus (Zhu et al. 2015) and English Wikipedia, with an approximate total of 3,300M tokens. It also enables significant improvement in many downstream tasks with relatively minimal task-specific fine-tuning. To study the quality of mention-span representations, we extract mention-span embeddings from BERT-base (12-layer Transformer, 768-hidden) and BERT-large (24-layer transformer, 1024-hidden). We also want to compare the effect of fine-tuning BERT-based models on the quality of span representations, therefore we use two variant of BERT models: a) fine-tuned on coreference resolution task and b) without any fine-tuning.

# 5 Experiments

We attempt to investigate to what extent the proposed span representation using BERT embeddings and fine-tuned on corefence resolution task can encode coreference relations. We also want to answer whether if the span representation is able to encode long-range coreference phenomena effectively, or is it just simply modeling local coreference relation? Our experiments, which we describe below, are intended to analyze coreference information contained in the span representation used in the span-ranking model.

## 5.1 Dataset

For our experiments, we use the coreference resolution annotation from the CoNLL-2012 shared task based on the OntoNotes dataset (Pradhan et al. 2012). The dataset comprises of roughly one million words from various genre such as newswire, magazine articles, broadcast news, broadcast conversations, web data and conversational speech data, and the New Testament in English, which was splitted into 2802 training documents, 343 validation documents, and 348 test documents. On average, the training documents contain 454 words. The largest document contains a maximum of 4009 words. Beside entities, coreference is also annotated for mentions that refer to the same event (e.g. "The European economy **grew** rapidly over the past years, **this growth** helped raising...). The main evaluation of the dataset is the average F1 score of three metrics – $MUC$ (Vilain et al. 1995), $B^3$ (Bagga and Baldwin 1998) and $CEAF_\phi 4$ (Luo 2005). Since OntoNotes only provide annotations for positive examples, we generate our own negative examples according to the aforementioned method (§4.2). We also cast the original annotation provided in the dataset into JSON format in order to work with BERT-coref.

## 5.2 Implementation and Hyperparameters

We extend the original Tensorflow implementation of BERT-coref [2] in order to build our probing model with Keras frontend (Chollet 2015). Our probing model are trained for 50 epochs, using early stopping with patience of 3 and batch size of 512. For optimization, we use Adam (Kingma and Ba 2015) with a learning rate of 0.001. The weights of the probing model is initialized with Kaiming initialization (K. He et al. 2015) and the size of the hidden layer is $d = 1024$ with rectified linear units (Nair and Hinton 2010).

As mentioned previously, we use both original pre-trained BERT model without fine-tuning the encoder weights and BERT model that has been fine-tuned on coreference resolution task (i.e. on OntoNotes annotations). For fine-tuned BERT model, we take the models that yield the best performance on (Joshi et al. 2019), which were trained using 128 word-pieces for BERT-base and 384 word-pieces for BERT-large. The fine-tuned model was trained using splitted OntoNotes documents where each segment non-overlaps and fed as a separate instance. This is done as BERT can only accept sequences of at most 512 word-pieces and typically OntoNotes documents require multiple segments to be read entirely. In all of our experiments, we use the cased English BERT models.

## 5.3 Baseline

As our baseline, we use span representations introduced in the edge probing framework (Tenney et al. 2019). First of all, we take concatenated contextual embeddings for pair of mention-span

[2] https://github.com/mandarjoshi90/coref

$e^{(1)} = [x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, ..., x_n^{(1)}]$ and $e^{(2)} = [x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, ..., x_n^{(2)}]$ as inputs. We then project the concatenated contextual embeddings $e^{(1)}$ and $e^{(2)}$ to improve performance:

$$e^{(i)} = Ae^{(i)} + b \tag{16}$$

where $i = (1, 2)$, $A$ and $b$ are weights of the projection layer. The parameters in the projection layer are shared between representations $e^{(1)}$ and $e^{(2)}$. Afterwards, we apply self-attentional pooling operator in (§4.1) over the projected representations to yield a fixed-length span representations. This also helps to model head words for each mention-span. These mention-span representations are then concatenated together and passed to the probing model to predict whether they corefer or not. It is important to note that the self-attention pooling is computed only using tokens within the boundary of the span. As a result, the model can only access information about the context surrounding the mention span through the contextual embeddings. We take the contextual embeddings from activations of non-finetuned BERT final layer, while freezing the encoder. The baseline probing architecture is depicted in Figure 2.

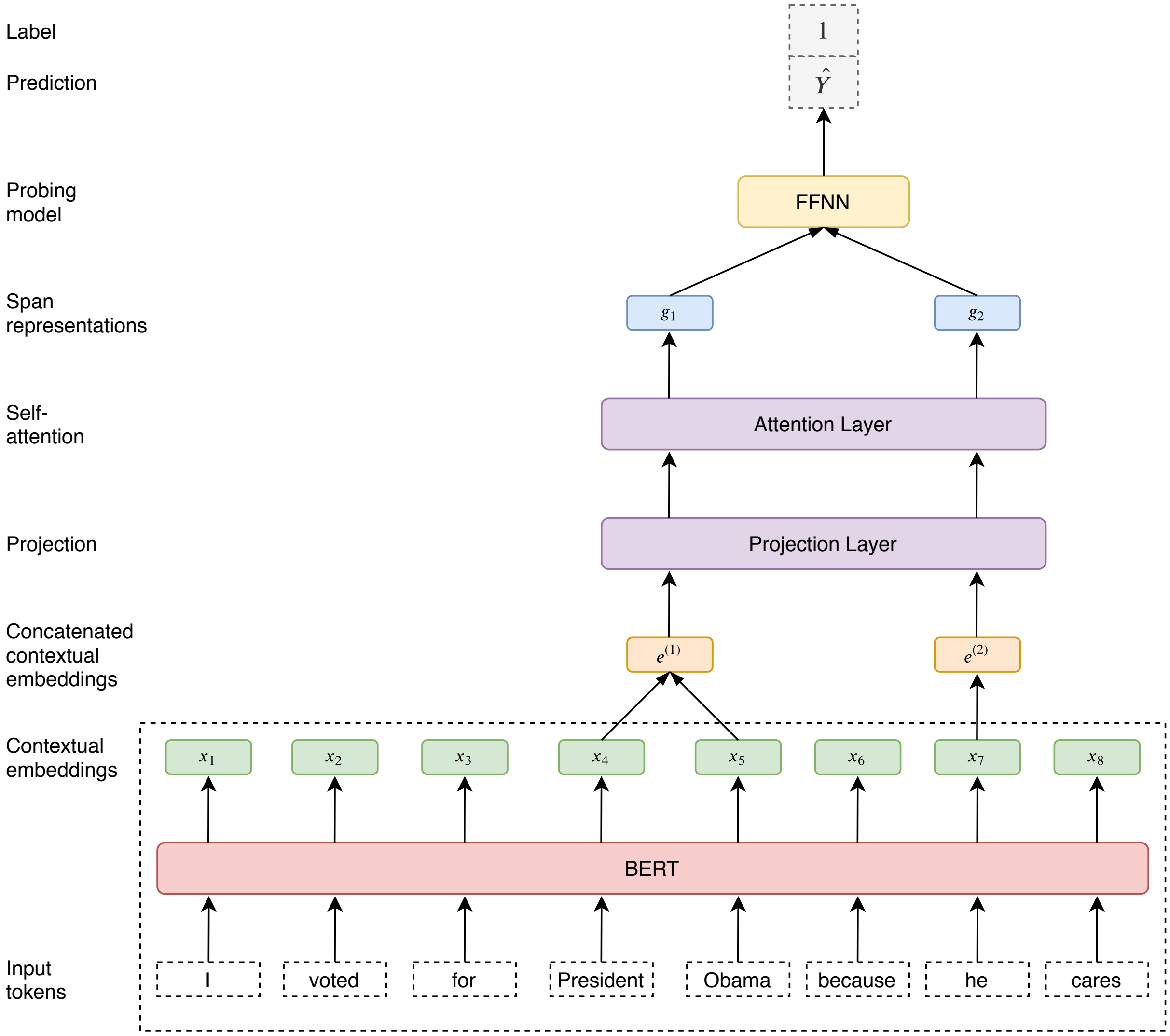


Figure 2: Probing architecture for baseline span representation. All parameters inside the dashed lines are frozen, while we train the feed-forward neural network to extract coreference information from the concatenated contextual embeddings $e^{(1)}$ and $e^{(2)}$. We used shared weights for both projection and self-attentional layer so that the model can learn the similarity between representation of mention-spans.

We compare the span representation used in the span-ranking model against the baseline, as it measures the performance that the probing model can achieve from the representation that is constructed from lexical priors alone, without any access to the local context within the mention-span. The resulting span representation have a dimension of $d = 768$ for BERT-base and $d = 1024$ for BERT-large.

### 5.4 Long-range Coreference

We want to understand better about the capability of the span representation from BERT encoder that is fine-tuned on coreference resolution task. Is it able to encode long-range coreference, or is it just simply modelling local coreference? We also want to answer how well does the model learn from local and long-range context within the mention-span. In order to answer both of these questions, we extend our baseline by introducing a convolutional layer to incorporate surrounding context and improves the baseline span representation. We replace the projection layer in our probing architecture with a fully-connected 1D CNN layer with a kernel width of 3 and 5, stride of 1 and same padding to properly include contextual embeddings at the beginning and end of the mention-span. This is equivalent to seeing ±1 and ±3 tokens around the center word respectively. We also initialize the weights of the CNN layer with Kaiming initialization (K. He et al. 2015). Using this extended probing architecture with CNN layer as another baseline, which we will refer to as CNN baseline in this paper, enables us to see the contribution of local and non-local context to the performance of the probing model.

To investigate whether if the span representation is able to capture long-range coreference relation or not, we see how our probing model performs with various distance between mention-spans. We separate pair of mention-spans that appear in the OntoNotes test set into several buckets, based on the distance between last token of mention-span $w_i$ and first token of mention-span $w_j$, where $w_j$ occurs after $w_i$. Each bucket contains at least 50 examples of pair of mention-spans.

## 6 Results

### 6.1 Comparison of Probing Models

We report the accuracies and F1 scores of our probing model with span representations constructed with BERT encoder. Table 1 compares the performance of probing model using span representations finetuned on OntoNotes dataset against baseline span representation and CNN baseline that utilize non-finetuned BERT encoder. Using a simple linear probing model, we demonstrate that span representation in BERT-coref encodes a significant amount of coreference information, as we are able to train the probing model to predict whether a pair of mention spans corefer or not based on their span representations alone. Both BERT-base c2f and BERT-large c2f consistently scores above 90% (accuracy and F1 score) on OntoNotes test set.

We observe that both BERT-base c2f and BERT-large c2f performs better in predicting coreference arc between a pair of mention-spans compared to their respective baseline (by 2.37 points for accuracy and 2.18 F1 points on average). We find that although training the contextual probing model to learn contextual features for coreference arc prediction helps in encoding the necessary coreference information into the baseline span representation, it still cannot outperform the probing model that utilizes span representation in BERT-coref. This might be caused by better coreference-related features that is learned by BERT encoder when it is fine-tuned on OntoNotes.

We find that fine-tuning the span representation on coreference resolution task helps to encode local and long-range context inside the mention-span efficiently. This can be observed from the performance of CNN baseline, where the probing model is trained using 1D CNN layer with kernel

| | Accuracy | F1 Score |
|---|---|---|
| BERT-base c2f (cased, fine-tuned) | 92.93 | 93.02 |
| BERT-large c2f (cased, fine-tuned) | $93.65^*$ | $93.68^*$ |
| BERT-base CNN (cased, original, kernel=3) | 89.51 | 89.91 |
| BERT-base CNN (cased, original, kernel=5) | 89.04 | 89.28 |
| BERT-large CNN (cased, original, kernel=3) | 90.27 | 90.35 |
| BERT-large CNN (cased, original, kernel=5) | 88.09 | 88.28 |
| BERT-base (cased, original) baseline | 90.37 | 90.65 |
| BERT-large (cased, original) baseline | 91.47 | 91.69 |

Table 1: Comparison of the probing model's performance with various mention-span representations evaluated on OntoNotes test set. Asterisk denotes the best performance on each metric. BERT-large c2f improves the accuracy and F1 score over the probing baseline by 3.28% and 3.03% for the base variant, while for BERT-large baseline, the improvements are 2.18% and 1.99% respectively.

width of 3 and 5 to allow us to see contribution of local and long-range dependencies, but ultimately still perform lower compared to BERT-coref. Surprisingly, our baseline span representation which is constructed from only lexical prior performs better compared to the CNN baseline span representation on both metrics. We attribute this due to our decision of taking contextual embeddings from the final layer of pre-trained BERT, as most transferrable representations from contextual encoders trained with language modeling objective tend to occur in the intermediate layers, and that the topmost layers might be overly specialized for next-word prediction (Liu et al. 2019; Peters, Neumann, Iyyer, et al. 2018; Peters, Neumann, Zettlemoyer, et al. 2018; Blevins, Levy, and Zettlemoyer 2018; Devlin et al. 2019). This may cause the CNN layer to learn suboptimal representations from the mention-span.

### 6.2 Ablations

To find out the importance of each component in BERT-coref span representations, we do ablation study on each part of the representation and report the accuracy and F1 score of the probing model on the test set of the data, which is depicted on Table 2. The head-finding attention mechanism is crucial for coreference-arc prediction, as it contributes the highest to the final result with 0.98 and 0.95 points for accuracy and F1 score on average, respectively. This is consistent with previous findings from (Lee, L. He, Lewis, et al. 2017), where attention mechanism is shown to be able to learn representation that is important for coreference. We also observe that span-width embeddings play an important role in determining coreference relation, seeing that without them the performance degrades on average by 0.4 and 0.37 for accuracy and F1. Contrary to the head-finding attention and span-width embeddings, boundary representations did not contribute much to the model's performance. We hypothesize that although boundary representations encode rich amount of information for coreference resolution, it is not significant for coreference arc prediction, as the model does not have to predict distribution over possible spans.

### 6.3 Encoding Long-range Coreference

We compare how our probing model performs with various span separation distance in order to explore whether if the span-representation is able to encode long-range coreference or not. Figure 3 depicts F1 score as a function of the distance of wordpiece tokens between a pair of mention-spans. Although performance with BERT models degrade with larger distance over pair of mention-span, the span representation in BERT-coref holds up better in general compared to the baseline or CNN

| | Accuracy | F1 Score | ΔAccuracy | ΔF1 Score |
|---|---|---|---|---|
| BERT-base c2f (cased, fine-tuned) | $92.93$ | $93.02$ | | |
| - boundary representations | $92.88$ | $92.96$ | $-0.05$ | $-0.06$ |
| - head-finding attention | $92.05$ | $92.16$ | $-0.88$ | $-0.86$ |
| - span-width embeddings | $92.46$ | $92.56$ | $-0.47$ | $-0.46$ |
| BERT-large c2f (cased, fine-tuned) | $93.65$ | $93.68$ | | |
| - boundary representations | $93.47$ | $93.49$ | $-0.18$ | $-0.19$ |
| - head-finding attention | $92.57$ | $92.65$ | $-1.08$ | $-1.03$ |
| - span-width embeddings | $93.32$ | $93.41$ | $-0.33$ | $-0.27$ |

Table 2: Comparison of the probing model on OntoNotes test set with various components removed. The head-finding attention and span-width embeddings contribute significantly to the performance of the probing model.

baseline. The BERT-base variant experiences minor degradation in performance to 5 points when $d = 125$ tokens, while for BERT-large, the F1 score dropped only by 7 points between $d = 0$ tokens and $d = 250$ tokens, which suggests that the depth of Transformer layer helps to encode long-range coreference. However, we lack sufficient evidence to suggest that the span representation is able to encode long-range coreference relation efficiently, seeing that although already the encoder has been fine-tuned on OntoNotes, the model still cannot perform consistently across distant spans, with the lowest F1 score of 0.67 and 0.75 points for BERT-base and BERT-large respectively, when $d = 451$ to $475$ tokens.

# 7 Discussion

## 7.1 Design of Qualitative Analysis

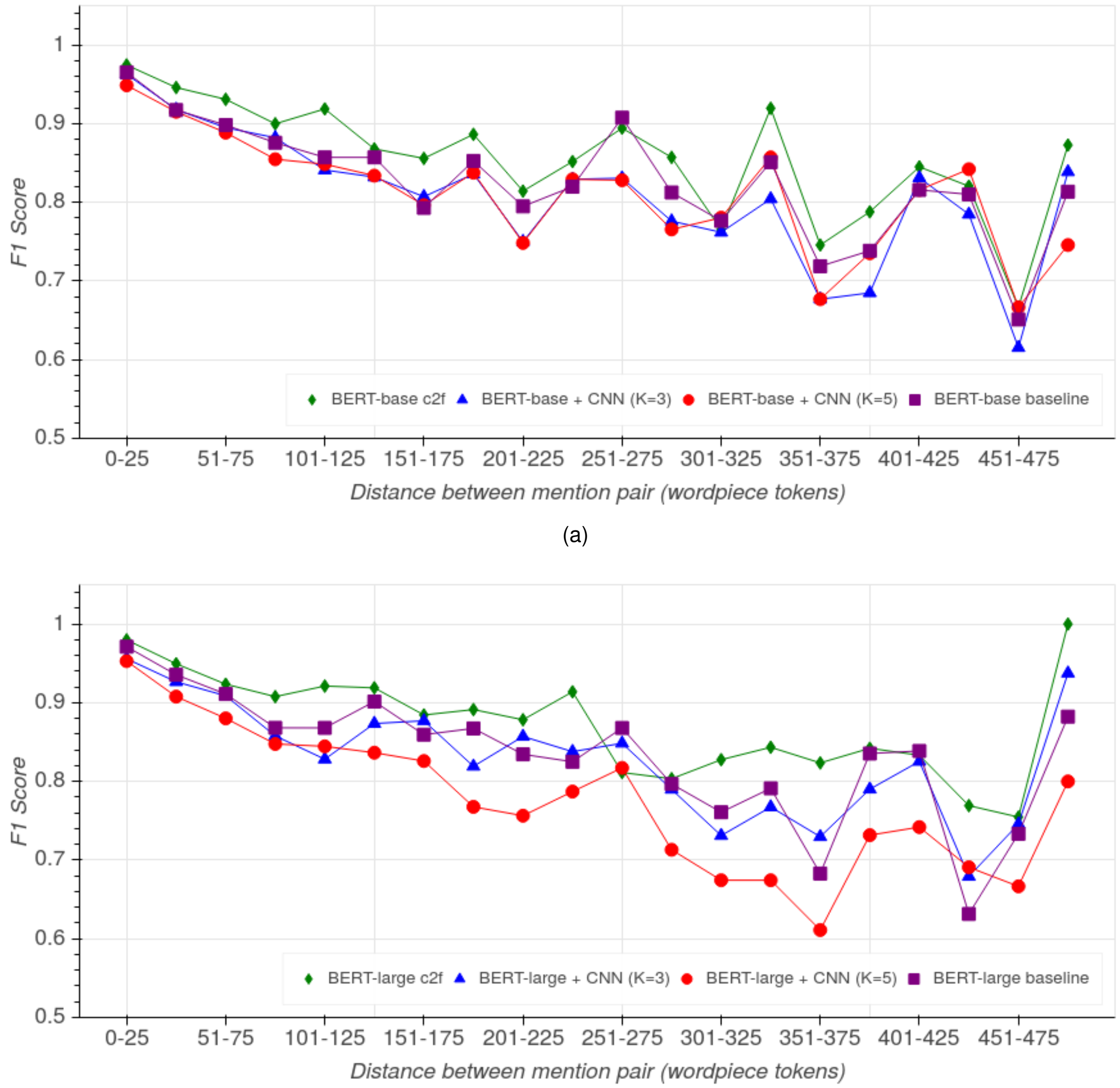
Our aim is to present qualitative error analysis for predicted coreference between mention-pairs. We identify and characterize a set of challenging errors common to state-of-the-art systems dealing with coreference resolution in English. Such quialitative analysis enables us to detect patterns in errors that our system produces and to discuss underlying reasons for such errors. As true positives we treat the pair of one mention and another mention belonging to the same coreference chain which matches the corresponding mention in the gold standard.

We conducted our error analysis on 4 models: BERT-base c2f (cased, fine-tuned), BERT-large c2f (cased, fine-tuned), BERT-base (cased, original) baseline, and BERT-large (cased, original) baseline. We used a subsample of 100 coreference clusters from the Ontonotes test dataset. We divide errors into the following categories: *Related Entities, Lexical, Pronouns, Gender, Mention Paraphrasing, Conversation,* and *Misc.* Although *Gender* can be considered a subcategory of *Pronouns*, the decision to separate it is motivated by the curiosity to find out whether gender bias is present in the models.

In the table: bold are false positives, cursive are false negatives.

Our hypothesis is that mentions that are separated by a large distance will have a higher error rate in all categories.

The most difficult for all models is ... add this. There is a tendency for all systems to ... add this. The least problematic category for all models is...



(a)

(b)

Figure 3: F1 score of probing model as a function of separating distance between two mention-spans with BERT-base (3a) and BERT-large (3b) on test set. The performance with either BERT-base or BERT-large tend to fall off as distance between wordpiece token increases.

### 7.2 Anaphora Resolution

Pronouns coreference spanning over 100 tokens is intrinsically difficult, even for a human. The model predicts pronouns relating to a mention 200 tokens in distance.

Agreement is good even when predicting oven 200 words. - Them == her colleagues guesses correctly

### 7.3 Distinct Entities

Model merges distinct entities into one cluster

## 7.4 Mention Paraphrasing

Error in failing to label two mentions as referring to the same entity on one hand and labelling two distinct mentions as coreferent on the other hand.

Goods and Intellectual property Rights are deemed coreferent by joshi base: legally speaking different things. Might occur in similar laws.

Jesus and Young Man. Part of the dialog between Jesus and a young man.

## 7.5 Morphologically Related

The word forms of the two mentions are similar, or the same graphems appear in both mentions, but they refer to different entities.

## 7.6 Gender

Usually in conext of a gender pronoun relating to a name of a person. no problems with common names. Troeble with Scooter Libby as she.

## 7.7 Temporal and Spacial Agreement

Mention paraphrasing with reference to a specific date (e.g. January 3 and yesterday) and a specific spacial areas (two Chinese provinces).

1996 and 1997 are labeled as coreferent by joshi's base model. Or this year and 1994. This is a mistake that a human will likely not do.

## 7.8 Miscellaneous

Majority of mentions here involve a large Noun Phrase that is falsely identified as coreferent with a verb or a noun. Or anything else that didn't generalise onto other examples.

The most difficult for

## 7.9 Other Languages

The results we introduced in this paper has been tested solely on an English corpus. Hence, the high results may be a phenomena particular to this language. For morphologically or syntactically complex languages the models would have struggled more. A benefit of using neural models for coreference resolution is their ability to use contextualized word embeddings to capture similarity between words, a property that many traditional feature-based models lack. However, a similar analysis should be done for other languages before one can generalize that machine-learning based models outperform rule-based models universally.

### 7.10 Future Work

- everything we don't have the time for
- mention other corpora that could be used for finetuning (Winogrande, GAP...)
- apply roberta, spanbert (sota result, architecture still use c2f)
- try building span representations from intermediate layers, use scalar mixing
- comparing attention head of Joshi's and baseline?

## 8 Conclusion

- so what have we learned about coref in general and local dependencies in particular

american/british english consistency oxford comma?

## References

Bagga, Amit and Breck Baldwin (1998). “Algorithms for Scoring Coreference Chains”. In: *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pp. 563–566.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1409.0473.

Bengtson, Eric and Dan Roth (2008). “Understanding the Value of Features for Coreference Resolution”. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 294–303. URL: https://www.aclweb.org/anthology/D08-1031.

Blevins, Terra, Omer Levy, and Luke Zettlemoyer (2018). “Deep RNNs Encode Soft Hierarchical Syntax”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 14–19. DOI: 10.18653/v1/P18-2003. URL: https://www.aclweb.org/anthology/P18-2003.

Chollet, François et al. (2015). *Keras*. https://keras.io.

Clark, Kevin and Christopher D. Manning (2016). “Deep Reinforcement Learning for Mention-Ranking Coreference Models”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2256–2262. DOI: 10.18653/v1/D16-1245. URL: https://www.aclweb.org/anthology/D16-1245.

Daniel Jurafsky, James H. Martin (2019). *Speech and Language Processing. Draft of October2, 2019*. URL: https://web.stanford.edu/~jurafsky/slp3.

Devlin, Jacob et al. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

Durrett, Greg and Dan Klein (2013). “Easy Victories and Uphill Battles in Coreference Resolution”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1971–1982. URL: https://www.aclweb.org/anthology/D13-1203.

Fernandes, Eraldo, Cicero dos Santos, and Roy Milidiu (2012). “Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution”. In: *Joint Conference on EMNLP and CoNLL - Shared Task*. Jeju Island, Korea: Association for Computational Linguistics, pp. 41–48. URL: https://www.aclweb.org/anthology/W12-4502.

Haghighi, Aria and Dan Klein (2009). “Simple Coreference Resolution with Rich Syntactic and Semantic Features”. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*. EMNLP 09. Singapore: Association for Computational Linguistics, pp. 1152–1161. ISBN: 9781932432633.

Haghighi, Aria and Dan Klein (2010). “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*. Los Angeles, California: Association for Computational Linguistics, pp. 385–393. URL: https://www.aclweb.org/anthology/N10-1061.

He, Kaiming et al. (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, pp. 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123. URL: https://doi.org/10.1109/ICCV.2015.123.

Hobbs, Jerry R. (1978). “Resolving pronoun references”. In: *Lingua* 44.4, pp. 311–338. ISSN: 0024-3841. DOI: https://doi.org/10.1016/0024-3841(78)90006-2.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Iida, Ryu et al. (2003). “Incorporating contextual cues in trainable models for coreference resolution”. In: *In Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*. Citeseer.

Joshi, Mandar et al. (2019). “BERT for Coreference Resolution: Baselines and Analysis”. In: *Empirical Methods in Natural Language Processing (EMNLP)*.

Kádár, Akos, Grzegorz Chrupała, and Afra Alishahi (2017). “Representation of linguistic form and function in recurrent neural networks”. In: *Computational Linguistics* 43.4, pp. 761–780.

Kennedy, C. and B.K. Boguraev (1996). “Anaphora for everyone: Pronomial anaphora resolution without a paser.” In: pp. 113–118.

Kingma, Diederik P. and Jimmy Ba (2015). “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

Lappin, S. and H. Leass (1994). “An algorithm for pronominal anaphora resolution”. In: vol. 20(4), pp. 535–561.

Lee, Kenton, Luheng He, Mike Lewis, et al. (2017). “End-to-end Neural Coreference Resolution”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 188–197. DOI: 10.18653/v1/D17-1018. URL: https://www.aclweb.org/anthology/D17-1018.

Lee, Kenton, Luheng He, and Luke Zettlemoyer (2018). “Higher-Order Coreference Resolution with Coarse-to-Fine Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 687–692. DOI: 10.18653/v1/N18-2108. URL: https://www.aclweb.org/anthology/N18-2108.

Liu, Nelson F. et al. (2019). “Linguistic Knowledge and Transferability of Contextual Representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1073–1094. DOI: 10.18653/v1/N19-1112. URL: https://www.aclweb.org/anthology/N19-1112.

Luo, Xiaoqiang (2005). “On Coreference Resolution Performance Metrics”. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 25–32. URL: https://www.aclweb.org/anthology/H05-1004.

Martschat, Sebastian and Michael Strube (2015). “Latent Structures for Coreference Resolution”. In: *Transactions of the Association for Computational Linguistics* 3, pp. 405–418. DOI: 10.1162/tacl_a_00147. URL: https://www.aclweb.org/anthology/Q15-1029.

Nair, Vinod and Geoffrey E. Hinton (2010). “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML’10. Haifa, Israel: Omnipress, pp. 807–814. ISBN: 9781605589077.

Ng, Vincent (2005). “Supervised Ranking for Pronoun Resolution: Some Recent Improvements”. In: *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*. AAAI’05. Pittsburgh, Pennsylvania: AAAI Press, pp. 1081–1086. ISBN: 157735236x.

Ng, Vincent and Claire Cardie (2002). “Identifying Anaphoric and Non-Anaphoric Noun Phrases to Improve Coreference Resolution”. In: *COLING 2002: The 19th International Conference on Computational Linguistics*. URL: https://www.aclweb.org/anthology/C02-1139.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Peters, Matthew, Mark Neumann, Mohit Iyyer, et al. (2018). “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association* *for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202.

Peters, Matthew, Mark Neumann, Luke Zettlemoyer, et al. (2018). “Dissecting Contextual Word Embeddings: Architecture and Representation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1499–1509. DOI: 10.18653/v1/D18-1179. URL: https://www.aclweb.org/anthology/D18-1179.

Pradhan, Sameer et al. (2012). “CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes”. In: *Joint Conference on EMNLP and CoNLL - Shared Task*. Jeju Island, Korea: Association for Computational Linguistics, pp. 1–40.

Shi, Xing, Inkit Padhi, and Kevin Knight (2016). “Does String-Based Neural MT Learn Source Syntax?” In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1526–1534. DOI: 10.18653/v1/D16-1159. URL: https://www.aclweb.org/anthology/D16-1159.

Tenney, Ian et al. (2019). “What do you learn from context? Probing for sentence structure in contextualized word representations”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SJzSgnRcKX.

Turian, Joseph, Lev-Arie Ratinov, and Yoshua Bengio (2010). “Word Representations: A Simple and General Method for Semi-Supervised Learning”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 384–394. URL: https://www.aclweb.org/anthology/P10-1040.

Vaswani, Ashish et al. (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Vilain, Marc et al. (1995). “A Model-Theoretic Coreference Scoring Scheme”. In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. URL: https://www.aclweb.org/anthology/M95-1005.

Winograd, T. (1972). *Understanding Natural Language*. Academic Press.

Wiseman, Sam, Alexander M. Rush, and Stuart M. Shieber (2016). “Learning Global Features for Coreference Resolution”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 994–1004. DOI: 10.18653/v1/N16-1114. URL: https://www.aclweb.org/anthology/N16-1114.

Wiseman, Sam, Alexander M. Rush, Stuart Shieber, et al. (2015). “Learning Anaphoricity and Antecedent Ranking Features for Coreference Resolution”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1416–1426. DOI: 10.3115/v1/P15-1137. URL: https://www.aclweb.org/anthology/P15-1137.

Woods, William A., Ronald M. Kaplan, and B. L. Nash-webber (1972). “The lunar sciences natural language information system: final report”. In:

Yang, Xiaofeng et al. (2003). “Coreference Resolution Using Competition Learning Approach”. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 176–183.

Zhou, Liang, Miruna Ticrea, and Eduard Hovy (2004). “Multi-Document Biography Summarization”. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 434–441.

Zhu, Yukun et al. (2015). “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, pp. 19–27. ISBN:

9781467383912. DOI: 10.1109/ICCV.2015.11. URL: https://doi.org/10.1109/ICCV.2015.11.