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 State of the Art Coreference Models

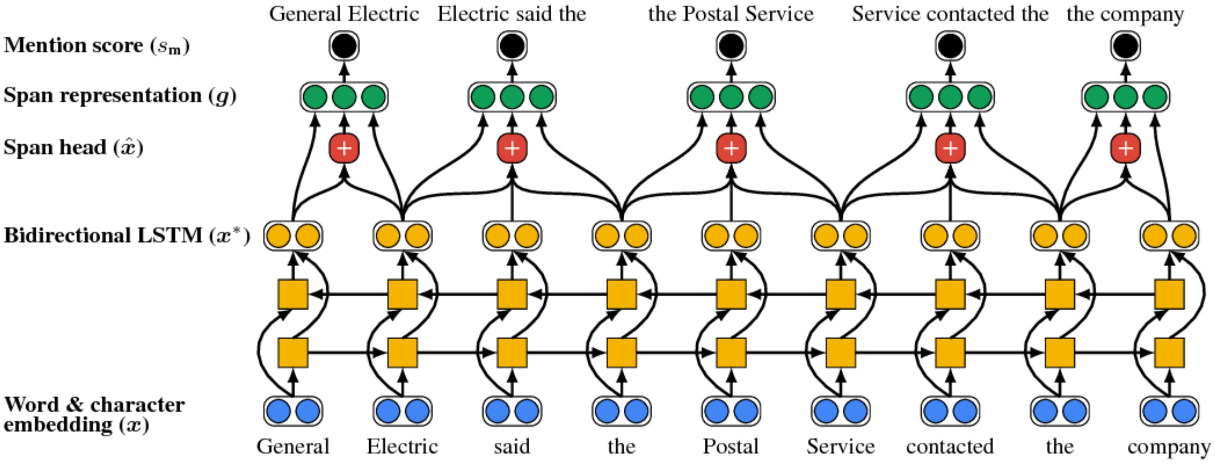


Figure 1: Figure 1: First step of the end-to-end coreference resolution model, which computes embedding representations of spans for scoring potential entity mentions. Low-scoring spans are pruned, so that only a manageable number of spans is considered for coreference decisions. In general, the model considers all possible spans up to a maximum width, but we depict here only a small subset. (Lee, He, Lewis, et al. 2017)

Lee, He, Lewis, et al. (2017) have proposed an span-ranking approach, which the authors describe as most similar to mention ranking, with reasoning over a larger space by detecting mentions and predicting coreference jointly in one end-to-end model. We will further refer to the model as *e2e-coref*.

The authors showed that from the space of all possible spans their model implicitly learns to produce meaningful mention candidates. A head-finding attention mechanism also learns a task-specific preference for head words, which has a strong correlation with head-word definitions in rule-based systems.

Building on to of the span-ranking neural architechture in Lee, He, Lewis, et al. (2017), Lee, He, and Zettlemoyer (2018) proposed a model that captures higher order interactions between spans in predicted clusters. We will further refer to this model as *c2f-coref*. It also alleviates the additional computational cost of higher-order inference due to a coarse-to-fine approach, which allows the model to at first compute a rough sketch of likely antecedents, and only at a later stage apply a more exhaustive inference. Hence, the coarse-to-fine approach expands the set of coreference links that the model is capable of learning.

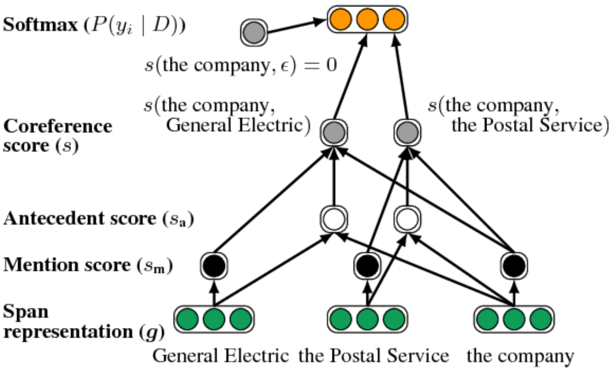

Softmax ($P(y_i \mid D)$)
$s(\text{the company}, \epsilon) = 0$
s(the company, General Electric)
s(the company, the Postal Service)
Coreference score (s)
Antecedent score (s_a)
Mention score (s_m)
Span representation (g)
General Electric
the Postal Service
the company

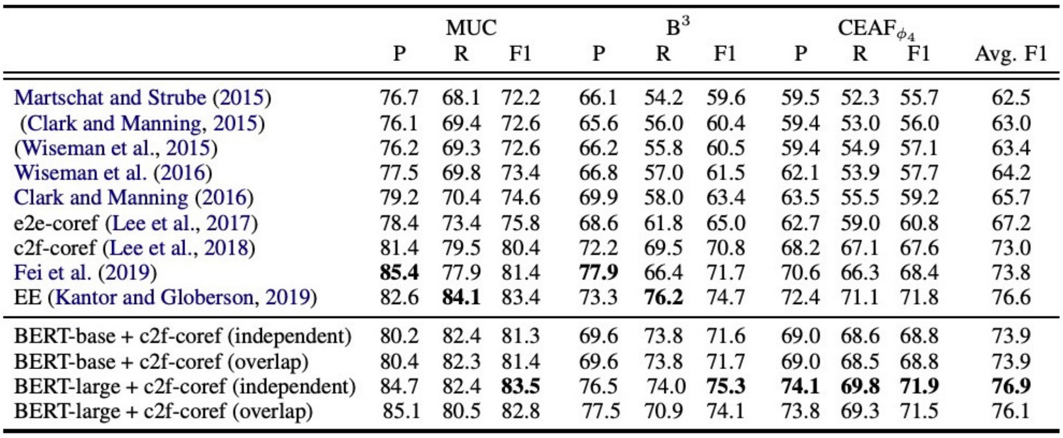
Figure 2: Second step of end-to-end model by Lee, He, Lewis, et al. (2017). Antecedent scores are computed from pairs of span representations. The final coreference score of a pair of spans is computed by summing the mention scores of both spans and their pairwise antecedent score.

To encode the span representations Lee, He, Lewis, et al. (2017) use bidirectional LSTM (Hochreiter and Schmidhuber 1997) to encode the lexical information of the inside and outside of each span. Lee, He, and Zettlemoyer (2018) additionaly use the newly published ELMo embedding representations by Peters et al. (2018) at the input to the LSTMs.

Joshi et al. (2019) took the architechture of Lee, He, and Zettlemoyer (2018) and improved the performence of the model further by replacing the LSTM-based encoder with the BERT transformer Devlin et al. (2019). We further refer to the model by Joshi et al. (2019) as *bert-coref*.

Devlin et al. (2019) made a significant break-through with their pre-trained BERT model. It can be finetuned with one additional output layer to create state-of-the-art models for a wide range of NLP tasks, such as question answering and language inference, without substantial taskspecific architecture modifications. BERT's training examples consist of 128 and 512 word pieces. Such passage-level training has been an important improvement over the previous methods. It helps the model learn dependencies over text sequences longer than one sentence, such as the previous state of the art model ELMo (Peters et al. 2018). Two model sized have been presented by Devlin et al. (2019): BERT-base(12 transformer blocks, hidden size 768, 12 self-attention heads, total Parameters=110M) and BERT-large (24 transformer blocks, hidden size 1024, 16 self-attention heads, total Parameters=340M).

Joshi et al. (2019) treat the first and last word-pieces in a mention (concatenated with the attended version of all word pieces in the span) as span representations. The researchers test both models with BERT-base and BERT-large transformers. With this change to the c2f-coref model the authors gain an additional 3.9% improvement on the OntoNotes compared to the already high results of Lee, He, and Zettlemoyer (2018). A quantitative analysis is shown in figure 3. A qualitative analysis done by the authors suggests that BERT-large (unlike BERT-base) is significantly better at distinguishing between related yet distinct entities or concepts.



	MUC			B^3			CEAF$_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Martschat and Strube (2015)	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
(Clark and Manning, 2015)	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
(Wiseman et al., 2015)	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
Wiseman et al. (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning (2016)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
e2e-coref (Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
c2f-coref (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Fei et al. (2019)	**85.4**	77.9	81.4	**77.9**	66.4	71.7	70.6	66.3	68.4	73.8
EE (Kantor and Globerson, 2019)	82.6	**84.1**	83.4	73.3	**76.2**	74.7	72.4	71.1	71.8	76.6
BERT-base + c2f-coref (independent)	80.2	82.4	81.3	69.6	73.8	71.6	69.0	68.6	68.8	73.9
BERT-base + c2f-coref (overlap)	80.4	82.3	81.4	69.6	73.8	71.7	69.0	68.5	68.8	73.9
BERT-large + c2f-coref (independent)	84.7	82.4	**83.5**	76.5	74.0	**75.3**	**74.1**	**69.8**	**71.9**	**76.9**
BERT-large + c2f-coref (overlap)	85.1	80.5	82.8	77.5	70.9	74.1	73.8	69.3	71.5	76.1

Figure 3: OntoNotes: BERT improves the c2f-coref model on English by 0.9% and 3.9% respectively for base and large variants. (Joshi et al. 2019)

2.3 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 End-to-End Span-ranking Model

4 Probing Mention-span Representations

4.1 Span Representation

Span representation plays an important role in span-ranking model (Lee, 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, 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 2015] over each word-pieces in the span:

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

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

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

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:

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

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

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 4.

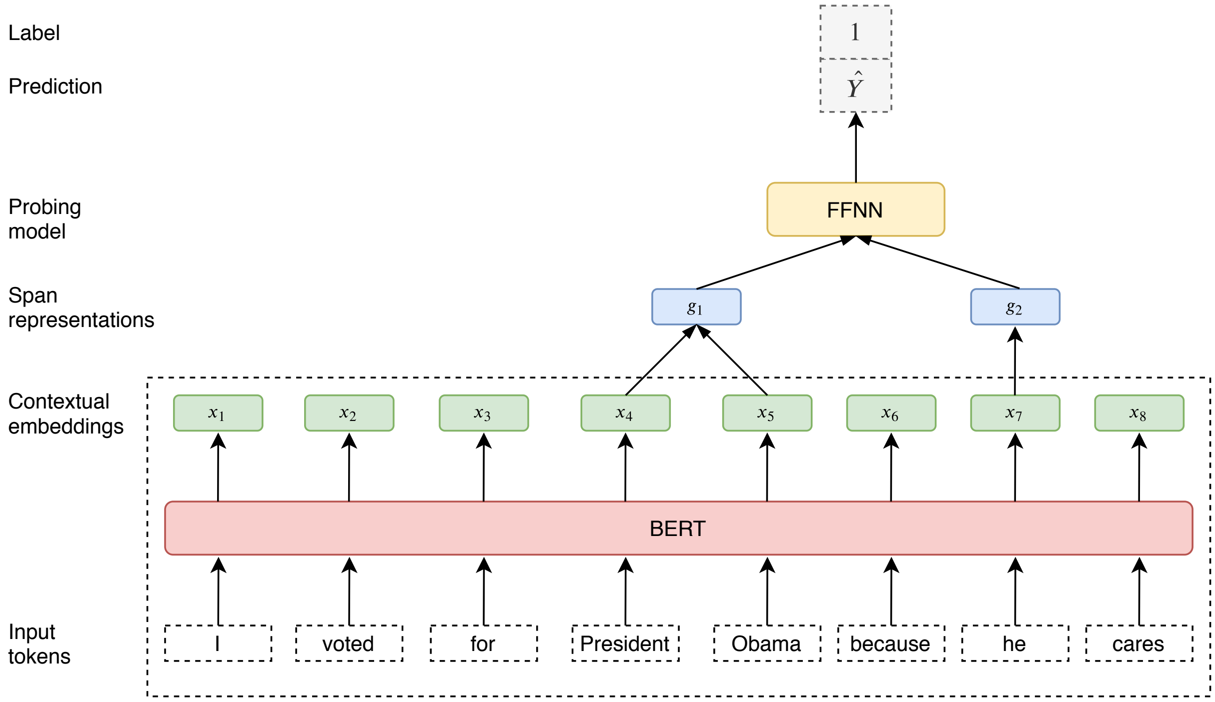


Figure 4: Probing architecture for span representation. The feed-forward neural network is trained to extract information from the span representations $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, while all the parameters inside the dashed line are frozen. The example depicts a mention-pair, where $\boldsymbol{g}_1$ corresponds to span representation of "President Obama", while $\boldsymbol{g}_2$ corresponds to "he". We predict $\hat{Y}$ as positive for this example.

We explore mention-span representations obtained from BERT. BERT (Devlin et al. 2019) is a language representations model which uses deep Transformer architecture [Vaswani 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 1995], B^3 [Bagga and Baldwin 1998] and $CEAF_{\phi}4$ [Luo 2015]. 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 [cite Keras]. 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] with a learning rate of 0.001. The weights of the probing model is initialized with Kaiming initialization [Kaiming] and the size of the hidden layer is $d = 1024$ with ReLU activation function.

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 $e^{(1)} = [x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, ..., x_n^{(1)}]$ 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:

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

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

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

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 5.

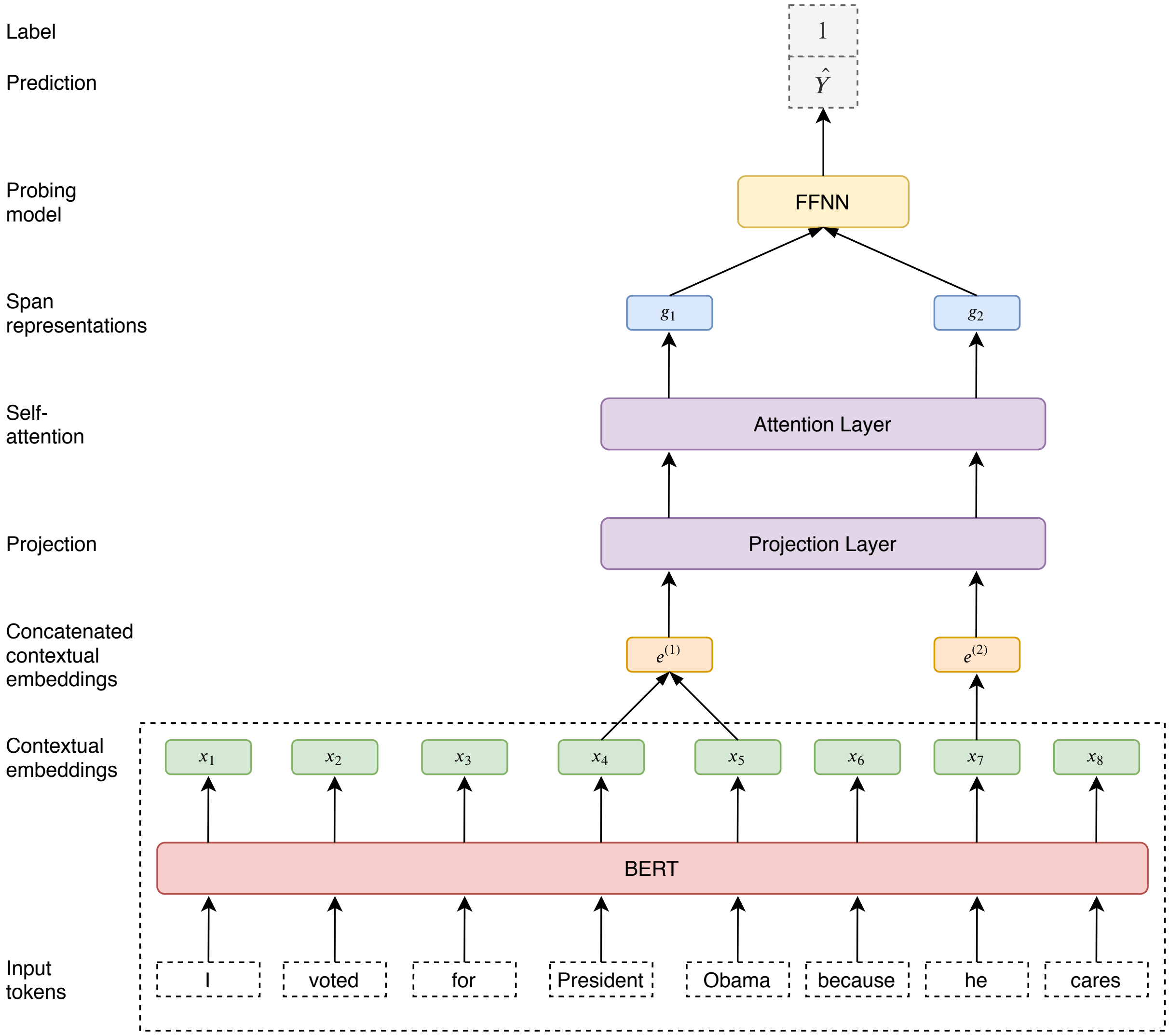


Figure 5: Probing architecture for baseline span representation. All parameters inside the dashed lines are frozen, while we train the feed-forward neural network to extract coreference information from the concatenated contextual embeddings $e^{(1)}$ and $e^{(2)}$. We used shared weights for both projection 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

6 Results

- some nice plots here
- some tables here

7 Discussion

- long long analysis of what we've seen
- generalisations, parallels
- what do this results tell us about coref and nlp in general
- discussion of why they are the way the are

We found that the model by Joshi et al. (2019) has performed very well. However, since the research is heavily based on the English language, it may be a phenomena particular to this language. The nominal declension in Ukrainian, for example, has seven cases; adjectives, pronouns have gender specific forms. Therefore, a similar analysis should be done for other languages before one can generalize the findings universally.

A benefit of using neural models for coreference resolution is their ability to use word embeddings to capture similarity between words, a property that many traditional feature-based models lack.

7.1 Future Work

- everything we don't have the time for
- mention other corpora that could be used for finetuning (Winogrande, GAP...)
- apply roberta, etc.

8 Conclusion

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

References

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.

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.

– (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.

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.

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.

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.

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.

Peters, Matthew E et al. (2018). “Deep contextualized word representations”. In: *Proceedings of NAACL-HLT*, pp. 2227–2237.

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.

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.