Coreference resolution for Slovene language: draft

Matej Klemen, Blažka Blatnik, Martin Čebular
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113, 1000 Ljubljana
{mk3141, bb3172, mc0239}@student.uni-lj.si

Abstract

In this paper, we present our work on coreference resolution for Slovene language using coref149 and SentiCoref datasets. We present coreference resolution as mention ranking problem and introduce three neural network models. First, we present the baseline model, a linear model with handcrafted features. Then, we introduce a non-contextual neural model using word2vec embeddings, and two contextual neural models using ELMo and BERT embeddings. We perform automated quantitative evaluation of the models using MUC, B3 and CEAFe scores as well as qualitative evaluation of the predictions. On the coref149 dataset, best CoNLL 2012 score was achieved by the contextual model with BERT embeddings (0.679), a result only slightly better than the one of the baseline (0.673). Generally, neural models performed relatively poorly on this dataset, which we attribute to small size of the dataset. On the other hand, the best performance on the SentiCoref dataset was achieved by the contextual model with ELMo embeddings and a context encoder, achieving a CoNLL 2012 score of 0.803.

1 Introduction

Coreference resolution is a task where the goal is to identify and group together all entity mentions that refer to a common entity in the text. Generally, the task can be thought of as a combination of mention detection and mention clustering and many approaches explicitly perform these two steps when doing coreference resolution. The mention detection step deals with the detection of all entities that refer to some entity in the text. Mention clustering then divides the entities into groups based on the entity they refer to.

In our work, we assume the mentions are already detected and focus only on mention clustering. Specifically, we focus on Slovene language, which has not yet been the subject of much research. For our experiments, we use two datasets: coref149 (Žitnik and Bajec, 2018), consisting of 149, and SentiCoref (Žitnik, 2019), consisting of 873 coreference-annotated documents. For our baseline, we implement a linear coreference resolution model that uses handcrafted features. We then try to improve its performance using neural network based approaches. Our goal is to see if the process of designing linguistic features, which requires a lot of domain knowledge, can be replaced by models leveraging various types of embeddings. Additionally, we want to see which linguistic properties these embedding based models are able to capture. We evaluate our approaches on the datasets using automated metrics and qualitatively compare their strengths and weaknesses. The source code for our experiments is available online [1].

The rest of the paper is structured as follows. In Section 2 we provide an overview of existing approaches to coreference resolution. In Section 3 we describe the formulation of mention clustering we use and the models we experiment with. In Section 4 we provide the results of our work, which we then discuss in Section 5. In Section 6 we summarize our work and provide some possible directions for further research.

2 Related Work

Most coreference resolution systems deal with two tasks: mention detection and mention clustering. Since our work is on mention clustering, we omit the literature on mention detection.

One approach to mention clustering is to treat it as a binary classification problem, where the goal is to determine whether two mentions are coreferent or not (Soon et al., 2001) (Ng and Cardie,

[1] https://github.com/matejklemen/slovene-coreference-resolution

2002). The problem of this approach is that it treats all coreference candidates independently, so it cannot choose the most probable candidate when multiple valid ones exist. A different way to do mention clustering, which solves this problem, is with mention ranking (Wiseman et al., 2015). In this approach, candidates for coreference are scored in some way and the best scoring candidate is proclaimed as the coreferent mention. The benefit of this approach is that it does not consider candidates in isolation, but rather in comparison to other mentions. An approach which takes this even further is the entity-mention approach (Wiseman et al., 2015). Here, the models are trained to determine whether the currently considered mention belongs to some preceding coreference cluster (Yang et al., 2004). In our work, we make use of mention ranking approach.

Recently, an end-to-end approach to coreference resolution was introduced (Lee et al., 2017), where the two steps are combined and learned together using deep neural networks. This approach considers all spans of tokens up to specific length as candidates for coreference. The spans are then scored in isolation and as mention pairs to produce a final coreference score, which is used in the span ranking coreference resolution framework.

The end-to-end approach was further researched and improved upon, for example by using more sophisticated contextualized embeddings (Joshi et al., 2019), but as is the case in most of the other areas in nature language processing, the research is mostly focused on the English language. Some examples of research done for other languages include a Lithuanian rule-based approach (Žitkus et al., 2019) and approaches for Polish (Nitoń et al., 2018) and Basque (Urbizu et al., 2019) that use neural networks.

3 Methods

3.1 Mention ranking formulation of the task

In all of our approaches we treat coreference resolution as a mention ranking problem. We are given a document with information about which spans of words (mentions) refer to the same entity. We move through the mentions in the order of their appearance in document. For every mention, we determine which preceding mention (antecedent) it is coreferent with. This is done by assigning a coreference compatibility score to all candidates and selecting the mention with the highest score among them as the coreferent mention. Figure 1 showcases an example of mention ranking algorithm.

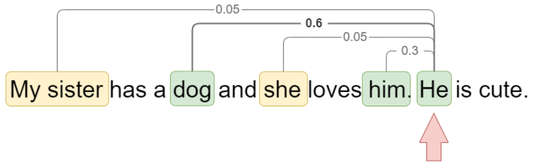


Figure 1: Mention ranking algorithm. Marked words represent mentions of two different entities, split by color, based on the entity they reference. Mention currently being processed is marked with a red arrow. We compute scores for all antecedent mentions. The mention with the highest score is selected as a co-reference.

The goal of our models is to make the coreference compatibility score high for coreferent mentions and low for non-coreferent mentions. Formally, our models minimize the mean cross-entropy between predicted and the ground truth antecedent probability distribution.

3.2 Baseline model

Our baseline model is a mention pair scorer based on linear regression and handcrafted features. Scores are obtained for every antecedent candidate appearing in the document and then normalized using softmax function. For constructing the features we require additional metadata such as part-of-speech tags and lemmas. For coref149, this metadata is provided in the ssj500k dataset, while for SentiCoref, this metadata is not provided, so we obtain it automatically with the Stanza library (Qi et al., 2020).

The features we use in our baseline model are based on already-proven ones, reported in existing literature (Žitnik et al., 2014). They are described in Table 1. Categorical features are encoded into binary ones using one-hot encoding.

3.3 Neural models

The neural architectures, which we experiment with, all reuse the same underlying neural network based scorer, originally introduced as part of an end-to-end system for coreference resolution (Lee et al., 2017). The scorer is shown in Figure 2.

For each mention in the pair, a three part mention representation is constructed. It is a concatenation of the features of the first token in the mention, the last token in the mention and the weighted combination of features for all mention tokens.

Both mention representations are concatenated together with their element-wise product. This represents the input that is then fed into a two hidden layer feedforward network.

The scorer is used in conjunction with different types of word embeddings: non-contextual and contextual. Embeddings we use are shown in Figure 3. For experiments with non-contextual embeddings, we use word2vec embeddings (Mikolov et al., 2013), which we provide in their original form on the input to the coreference scorer. For experiments with contextual embeddings, we use ELMo embeddings (Peters et al., 2018) and BERT embeddings (Devlin et al., 2019). Following the setup used by the authors of ELMo, we additionally encode the embedded document tokens using a bidirectional LSTM (Hochreiter and Schmidhuber, 1997), processing sentences independently. When using BERT, we follow the setup described in existing literature (Joshi et al., 2019), where BERT-embedded tokens are given as input to the coreference scorer. Because BERT has an effective maximum input length, we divide the longer documents into non-overlapping parts of maximum length and embed them independently. The embeddings we use correspond to the last hidden states of the BERT model. In order to perform batched coreference score computation, we pad the mentions to a fixed maximum span size. Mentions which are longer than the maximum size are truncated. The size is set in a way that most mentions do not get truncated, for example as the 95th percentile of all mention lengths. In all embedding-based models we use pretrained vectors (Kutuzov et al., 2017) (Ulčar and Robnik-Šikonja, 2019). Out of the three used types of embeddings, we only unfreeze the word2vec embeddings due to resource constraints and due to the small size of one of the used datasets.

4 Results

We split both datasets into a training, validation and test set in ratio 70%:15%:15%, fixing the random seed in the process. The validation set is used to select the best hyperparameters for our model as well as for regularization. The best model is selected with early stopping: once the validation loss does not decrease for five consecutive epochs, the training is stopped and the best state is used for evaluation.

For model evaluation MUC (Vilain et al., 1995), BCubed (Bagga and Baldwin, 1998) and CEAFe (Luo et al., 2004) scores are used. For each metric, three numbers are reported: precision, recall and F1 score. We compute the metrics using neleval[2] package. In addition to this, we also report the CoNLL 2012 score, which is the average F1 score of the three metrics and is intended to serve as a compact summary of the model's performance.

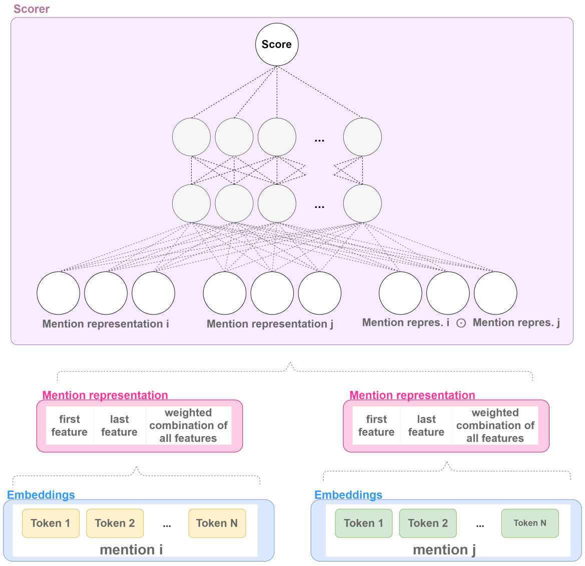


Figure 2: Neural coreference scorer architecture. The input to the coreference scorer are mention representations for both mentions and their element-wise product. Implemented models differ only in type of embedding used.

The results are shown in Table 2 and Table 3. Although there are some existing results for the coref149 dataset (Žitnik and Bajec, 2018), we do not include them as they are not directly comparable due to using different split proportions. Besides our baseline scorer and variations of a neural coreference scorer we also include results obtained by two trivial models, which show what kind of scores one can expect by default. The "Each-in-own" model puts each mention in its own cluster, while the "All-in-one" model puts all mentions of a document in single cluster.

Our baseline model achieves an average F1 score of 0.635 on SentiCoref and 0.673 on coref149. The best average F1 is achieved by the ELMo-based contextual scorer on SentiCoref (0.803) and by BERT-based contextual scorer on coref149 (0.679) dataset, though the latter improvements over the baseline are minor. Both best approaches have more balanced precision, recall

[2] https://github.com/wikilinks/neleval

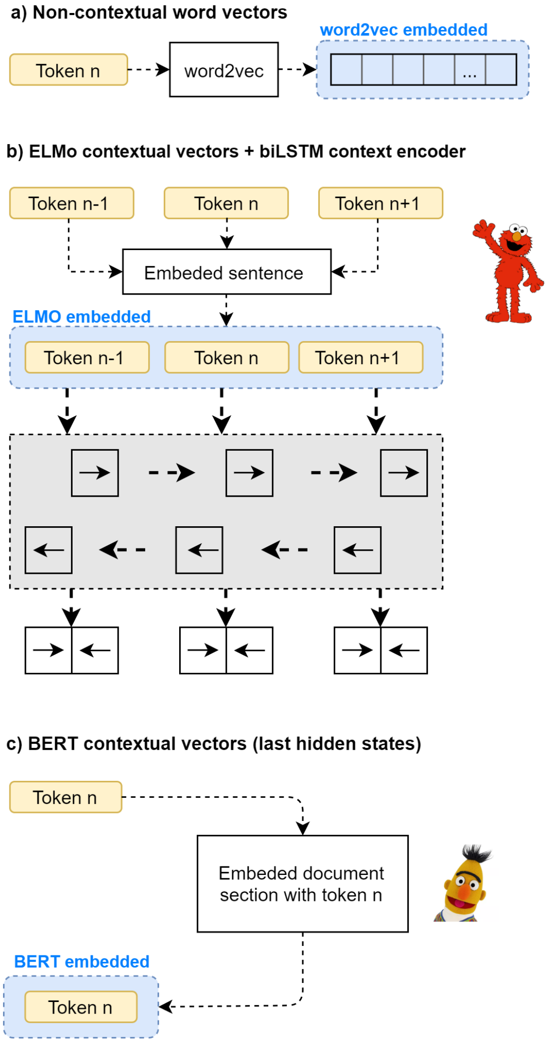

a) Non-contextual word vectors
Token n
word2vec
word2vec embedded
...
b) ELMo contextual vectors + biLSTM context encoder
Token n-1
Token n
Token n+1
Embeded sentence
ELMO embedded
Token n-1
Token n
Token n+1
c) BERT contextual vectors (last hidden states)
Token n
Embeded document section with token n
BERT embedded
Token n

Figure 3: Different embeddings used in our model: **a)** raw word2vec vectors, **b)** ELMo contextual vectors, additionally encoded with a bidirectional-LSTM and **c)** BERT contextual vectors.

Table 1: Features used in our baseline model.

Feature	Description
string_match	exact match for pronouns or match in lemmas
same_sentence	are both mentions in same sentence
same_gender	one-hot encoded vector for values: same gender, different gender, can't determine
same_number	one-hot encoded vector for values: match in number, don't match in number, can't determine
is_appositive	both mentions have noun-related tag and previous mention is followed by comma
is_alias	one mention is a subset of another
is_prefix	one mention is prefix of another
is_suffix	one mention is suffix of another
is_reflexive	one mention is followed by another that is reflexive pronoun
jaro_winkler_dist	similarity value between two mentions according to Jaro-Winkler metric

and F1 scores in comparison to the baseline approach, where one of precision or recall dominates the F1 score. The approach using non-contextual word embeddings achieves worse average F1 than our baseline on the coref149 dataset (0.544) and slightly better average F1 on SentiCoref (0.660). All of our approaches surpass the average F1 score achieved by the trivial models.

5 Discussion

The results on coref149 dataset indicate that using more complex models does not necessarily help with the performance. The ELMo based coreference scorer achieves worse results than our linear baseline, while the BERT based scorer achieves only slight improvement (which is not tested for statistical significance, so we can not claim it is better with absolute certainty). This is likely due to the small size of coref149 dataset (containing 149 documents). Although we freeze both BERT and ELMo, the ELMo based scorer has more trainable parameters due to an additional context encoder. Such a large amount of parameters can not reliably be set on a small dataset, resulting in worse performance than our baseline model. The opposite effect can be seen on SentiCoref, which is a much larger dataset (containing 873 documents) that enables learning more complex features. There, the additional context encoder helps ELMo based scorer achieve a substantial improvement in the average F1 score over the base-

Table 2: Results of our approaches on the **coref149 dataset**. For MUC, B3 and CEAFe, three metrics are reported: precision, recall and F1 score. ❄indicates that the underlying embeddings are frozen.

Model	MUC	B3	CEAFe	Avg. F1

Each-in-own	0.000;
	0.000;
	0.000
		1.000;
		0.531;
		0.673
			0.433;
			0.764;
			0.539
				0.404

All-in-one	0.507;
	1.000;
	0.648
		0.227;
		1.000;
		0.346
			0.481;
			0.095;
			0.152
				0.382

Baseline	**0.876**; 0.445; 0.560	**0.988**; 0.679; **0.787**	0.580; **0.856**; 0.674	0.673

Non-contextual neural (word2vec)	0.483;
	0.217;
	0.290
		0.905;
		0.616;
		0.723
			0.523;
			0.796;
			0.621
				0.544

Contextual neural (ELMo) ❄	0.458;
	0.378;
	0.403
		0.808;
		0.692;
		0.728
			0.629;
			0.737;
			0.663
				0.598

Contextual neural (BERT) ❄	0.588; **0.575**; **0.569**	0.789; **0.757**; 0.761	**0.714**; 0.732; **0.707**	**0.679**

Table 3: Results of our approaches on the **SentiCoref dataset**. For MUC, B3 and CEAFe, three metrics are reported: precision, recall and F1 score. ❄indicates that the underlying embeddings are frozen.

Model	MUC	B3	CEAFe	Avg. F1

Each-in-own	0.000;
	0.000;
	0.000
		1.000;
		0.372;
		0.525
			0.279;
			0.713;
			0.389
				0.305

All-in-one	0.641;
	1.000;
	0.770
		0.135;
		1.000;
		0.231
			0.384;
			0.027;
			0.050
				0.350

Baseline	**0.941**; 0.477; 0.617	**0.980**; 0.571; 0.707	0.462; **0.838**; 0.580	0.635

Non-contextual neural (word2vec)	0.740;
	0.581;
	0.642
		0.814;
		0.663;
		0.721
			0.546;
			0.741;
			0.617
				0.660

Contextual neural (ELMo) ❄	0.828; **0.825**; **0.824**	0.794; **0.802**; **0.793**	**0.797**; 0.793; **0.790**	**0.803**

Contextual neural (BERT) ❄	0.645;
	0.330;
	0.425
		0.826;
		0.464;
		0.582
			0.396;
			0.729;
			0.501
				0.503

line. The BERT based scorer only relies on its pretrained internal context encoding mechanism, which is trained on a different domain and not tuned in our experiments. The feedforward network alone does not seem to be able to capture the complex feature interactions required to learn coreference resolution.

Focusing on the MUC score in isolation, we observe that the baseline achieves a much higher precision than recall, while the best neural scorers for the datasets achieve balanced precision and recall scores. This means that the baseline rarely predicts coreference between mentions that are not actually coreferent, though it misses out on many coreferent mention pairs that exist in the documents. Similar phenomenon can be observed for B3 score. Most of the mentions in clusters, produced by our baseline, are put in correct clusters (achieving high precision), but the model misses many additional actual mentions (achieving low recall). For CEAFe, the opposite is true: the recall is higher than the precision for the baseline model.

Since the baseline model is a simple linear model, we are able to interpret what it manages to learn by looking at its weights. Figure 4 shows them for both datasets. Both from the weights as well as from observing the predictions of the model we notice that it puts a high emphasis on string equivalence and near-equivalence attributes. Concretely, the *string_match* and *is_suffix* attributes have high corresponding weights on both datasets. This is expected, since using same word or using close variations of the same word often implies reference to the same entity.

Upon observing the predicted coreference chains for the non-contextual model on SentiCoref, we notice that it is able to learn many of the properties that we explicitly have to encode in the baseline model. An example is shown in Figure 5, where we can see that the model learns string equality (matching the two “NKBM” mentions), partial string equality (matching “Minister Bajuk” and “Bajuk”) and some gender agreement related feature (knowing that “je pojasnil” is referring to a male person, in this case “Minister Bajuk”). Although the model is able to learn some linguistic properties, it is likely limited by the way the vocabulary is encoded. We notice that around quarter of all the mentions in SentiCoref (11144 out of 42738) contain at least one out-of-

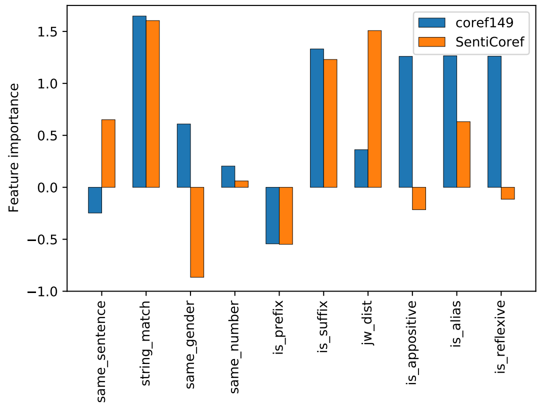
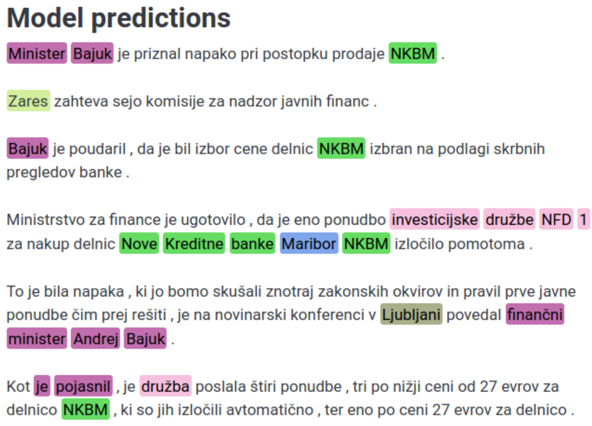


Figure 4: The weights of our linear coreference scorer. For both datasets, the model assigns high importance to the string equivalence based attributes.



Model predictions

Minister Bajuk je priznal napako pri postopku prodaje NKBM .

Zares zahteva sejo komisije za nadzor javnih financ .

Bajuk je poudaril , da je bil izbor cene delnic NKBM izbran na podlagi skrbnih pregledov banke .

Ministrstvo za finance je ugotovilo , da je eno ponudbo investicijske družbe NFD 1 za nakup delnic Nove Kreditne banke Maribor NKBM izločilo pomotoma .

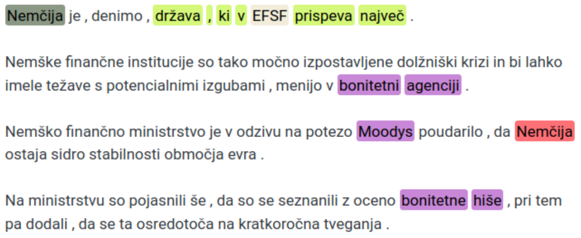
To je bila napaka , ki jo bomo skušali znotraj zakonskih okvirov in pravil prve javne ponudbe čim prej rešiti , je na novinarski konferenci v Ljubljani povedal finančni minister Andrej Bajuk .

Kot je pojasnil , je družba poslala štiri ponudbe , tri po nižji ceni od 27 evrov za delnico NKBM , ki so jih izločili avtomatično , ter eno po ceni 27 evrov za delnico .

Figure 5: An example of coreference clusters, predicted by the non-contextual neural model.

vocabulary token. These tokens get encoded with a vector that is common to all unknown words, which in a way introduces noise into the learning process and limits the model from possibly obtaining better results.

The results obtained by the ELMo based coreference scorer on SentiCoref as well as qualitative observation of predictions shows its consistency. For most documents it is able to correctly figure out most entities and its mentions. One of its limitations are sentences where two coreferent mentions, which are the first two mentions of an entity, are inside the same sentence. Such an example is shown in Figure 6. The context encoder only encodes the context inside same sentence, so the model fails when essential context is provided outside of the sentence.



Nemčija je , denimo , država , ki v EFSF prispeva največ .

Nemške finančne institucije so tako močno izpostavljene dolžniški krizi in bi lahko imele težave s potencialnimi izgubami , menijo v bonitetni agenciji .

Nemško finančno ministrstvo je v odzivu na potezo Moodys poudarilo , da Nemčija ostaja sidro stabilnosti območja evra .

Na ministrstvu so pojasnili še , da so se seznanili z oceno bonitetne hiše , pri tem pa dodali , da se ta osredotoča na kratkoročna tveganja .

Figure 6: An example where the ELMo based neural coreference scorer fails. In this example, “Nemčija” and “država, ki v EFSF prispeva največ” mention the same entity. The context encoder only encodes the sentence of the mention, so it misses out on useful information from the next few sentences, which could help in this case.

6 Conclusion

We experimented with and evaluated different mention clustering approaches for Slovenian language. We implemented a simple linear baseline using handcrafted linguistic features, then tried to see how well the embedding based models are able to automatically learn the features required for coreference resolution. We evaluated all the models on coref149 and SentiCoref datasets and achieved marginal improvements on coref149 using a BERT based scorer and substantial improvements (0.168 absolute CoNLL 2012) on SentiCoref using an ELMo based scorer. Our results show that all our approaches achieved non-trivial scores. Qualitative analysis showed that the neural coreference scorers are able to automatically learn some features which we had to explicitly encode in our baseline. The coref149 dataset proves to be a challenge to learn complex models on, likely due to its small size. SentiCoref does not have this problem since we are able to learn a complex ELMo based scorer on it, which achieves good results.

In the future, it would be interesting to explore the limits of the two coreference resolution datasets in more detail. More specifically, seeing how simple a neural network has to be in order to learn useful properties on the coref149 dataset. Conversely, on SentiCoref even more complex models could be tried since it is a big enough dataset. In addition to this idea, a possible future direction could be to check how well a subword based non-contextual embedding method performs. In our experiments, we noticed that many mention tokens get mapped to the “unknown” vector when using word2vec, which likely limits the learning capability of the model. Using subword based embeddings would get rid of this problem, as well as keep the model relatively simple.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

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

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 135. Association for Computational Linguistics.

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

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Bartłomiej Nitoń, Paweł Morawiecki, and Maciej Ogrodniczuk. 2018. Deep neural networks for coreference resolution for Polish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Matej Ulčar and Marko Robnik-Šikonja. 2019. High quality elmo embeddings for seven less-resourced languages. *arXiv preprint arXiv:1911.10049*.

Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 2019. Deep cross-lingual coreference resolution for less-resourced languages: The case of Basque. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 35–41, Minneapolis, USA. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 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)*, pages 1416–1426, Beijing, China. Association for Computational Linguistics.

Xiaofeng Yang, Jian Su, GuoDong Zhou, and Chew Lim Tan. 2004. An NP-cluster based approach to coreference resolution. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 226–232, Geneva, Switzerland. COLING.

Slavko Žitnik. 2019. Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0. Slovenian language resource repository CLARIN.SI.

Voldemaras Žitkus, Rita Butkienė, Rimantas Butleris, Rytis Maskeliunas, Robertas Damasevicius, and Marcin Woźniak. 2019. Minimalistic approach to coreference resolution in Lithuanian medical records. *Computational and Mathematical Methods in Medicine*, 2019:1–14.

Slavko Žitnik and Marko Bajec. 2018. Coreference resolution for Slovene on Annotated Data from coref149. *Slovenščina 2.0: empirical, applied and interdisciplinary research*, 6(1):37–67.

Slavko Žitnik, Lovro Šubelj, and Marko Bajec. 2014. Skipcor: Skip-mention coreference resolution using linear-chain conditional random fields. *PLOS ONE*, 9(6):1–14.