Computer Science and Information Systems 19(2):495–521 https://doi.org/10.2298/CSIS201120060K

Neural Coreference Resolution for Slovene Language

Matej Klemen and Slavko Žitnik

University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, 1000 Ljubljana
{matej.klemen, slavko.zitnik}@fri.uni-lj.si

Abstract. Coreference resolution systems aim to recognize and cluster together mentions of the same underlying entity. While there exist large amounts of research on broadly spoken languages such as English and Chinese, research on coreference in other languages is comparably scarce. In this work we first present SentiCoref 1.0 - a coreference resolution dataset for Slovene language that is comparable to English-based corpora. Further, we conduct a series of analyses using various complex models that range from simple linear models to current state-of-the-art deep neural coreference approaches leveraging pre-trained contextual embeddings. Apart from SentiCoref, we evaluate models also on a smaller coref149 Slovene dataset to justify the creation of a new corpus. We investigate robustness of the models using cross-domain data and data augmentations. Models using contextual embeddings achieve the best results - up to 0.92 average F_1 score for the SentiCoref dataset. Cross-domain experiments indicate that SentiCoref allows the models to learn more general patterns, which enables them to outperform models, learned on coref149 only.

Keywords: coreference resolution, Slovene language, neural networks, word embeddings.

1. Introduction

Coreference resolution is a task where the goal is to identify and group all entity mentions that refer to a common entity in the text. It is an important part of the attempt to understand language at a higher level and has its role across many other tasks in natural language processing. One such example is question answering, where the user can provide a complex query, often mentioning the same entity with different words to construct a less monotone sentence. For the system to determine what the user is asking and respond correctly, it must be able to figure out what the user is referring to across a long span of 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.

The most researched languages on this topic include broadly spoken languages such as English and Chinese. However, less-researched (and less-resourced) languages often possess interesting phenomena that do not appear in English and could provide a source

of difficulties for English systems. In our work, we focus on Slovene, an example of such a language, so far being the topic of few analyses.

We experiment with two datasets: coref149 and SentiCoref, with our work being the first analysis performed on the latter. As such, we provide a detailed description of the dataset and compare it with coref149 and some commonly used English coreference resolution datasets. We simplify our analysis, only studying the performance of systems on the mention clustering task and assuming that the system can do the mention detection step sufficiently accurately in advance. We analyze the performance of variously complex models on datasets providing a substantially different amount of resources to learn from. The studied models range in complexity from a simple linear baseline with features described by existing literature, to complex models that use contextual embeddings, pretrained on general multilingual or Slovenian data. Additionally, we study how transferable the patterns learned on both datasets are by either augmenting the datasets or learning a model on one dataset and evaluating its performance on the other. Throughout the analysis, we probe the effect of certain architectural decisions, such as embedding size or the amount of provided context, to additionally examine the capabilities of models and datasets. We complement the quantitative evaluation (using automated metrics) with additional qualitative analysis, outlining common mistakes in the best performing model. 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 datasets used in our experiments, with additional focus on SentiCoref. In Section 4 we describe the methods we use in our experiments. We then present and analyze the empirical results in Section 5. Finally, in Section 6, we summarize our work and provide some possible directions for further research.

2. Related Work

Coreference resolution is a widely studied problem in computational linguistics. Anaphoras and coreferent entities form a subset of discourse parsing [1] which is crucial for text understanding. A discourse is a group of interrelated sentences that contribute to a clear understanding only when read together. Anaphora on the other hand represents references (i.e. mentions) to items mentioned earlier in discourse. The primary anaphora type is the pronominal anaphora [2]. In contrast to anaphora, coreference identifies words or phrases (i.e. mentions) referring to an underlying unique entity. Most coreference resolution systems deal with two tasks: (a) mention detection and (b) mention clustering. As mention detection could be heuristically solved to identify mention candidates based on part-of-speech tagging, most systems focus on solving mention clustering. The latter is also the main focus of our work.

2.1. Coreference Resolution in English

Most approaches in coreference resolution transform the problem into a binary classification problem, where the goal is to determine whether two selected mentions are coref-

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

erent or not [3,4]. Prior to the use of deep learning approaches, methods based on conditional random fields [5] and rule-based methods [6] were achieving state-of-the-art results. The problem with such approaches is that they treat all coreference candidates independently, so they cannot choose the most probable candidate when multiple valid ones exist. Mention ranking was introduced as an improvement over those methods [7]. In these approaches, candidates for coreference are scored using a score and the best scoring candidates are selected as coreferent ones. The benefit of such an approach is that it does not consider candidates in isolation but jointly with other mentions. Another improvement is the entity-mention approach [7], where the models are trained to determine whether the observed mention belongs to one of the coreference clusters [8].

Recently, Lee et al. [9] introduced and evaluated the effectiveness of an end-to-end approach for coreference resolution, where the steps of mention detection and clustering are trained jointly using deep neural networks. They introduce a span ranking approach and optimize the two steps jointly by factoring the coreference compatibility score between two spans i and j into a part that models how likely it is that the two spans are actual mentions and a part that models how likely span j is an antecedent of span i. A potential problem of such an approach is that there are a lot of candidate spans to consider, which is solved by pruning the space of candidate spans. The method considers only a portion of the top N spans, selected based on the score that models how likely span i is a mention. In later work, the same authors [10] introduce another part to the coreference compatibility score that roughly models how likely span j is an antecedent of span i and use it to prune the candidate space even further. Subsequent work includes modifications such as the use of more sophisticated contextual embeddings [11] or more specialized ones [12] inside the end-to-end system. The latter work introduces SpanBERT, a modified version of Bidirectional Encoder Representations from Transformers (BERT) [13], which introduces a span masking and a span boundary objective as customized optimization objectives, designed to help span modeling tasks, such as coreference resolution.

2.2. Coreference Resolution in Non-English Languages

Due to the ubiquity of the English language and the availability of resources, the majority of work on coreference resolution is focused on the analysis of English data. However, studies exist for a wide variety of languages, presenting approaches that use rules, classic machine learning techniques or deep neural networks. Early approaches for various languages often tend to rely heavily on rules. Examples of such approaches include various systems in Polish [14], Lithuanian [15] and Russian [16]. These approaches offer a good starting point due to being well-studied and showing promising results in different languages. They are also relatively transparent, which enables their use in specific domains. For example, the Lithuanian approach performs coreference resolution on medical data.

After using rule-based systems, there was a shift towards using machine learning models combined with hand-engineered features. A positive aspect of such approaches is that there exist common features that work well across different languages, although they might have different importance. However, the features can automatically be weighted and combined by the models. This claim is supported by literature which adapts English systems and applies them to another language, such as the Polish adaptation [17] of Beautiful Anaphora Resolution Toolkit (BART) [18] as well as in the existing literature for Slovene language [19], where Žitnik and Bajec analyze the effectiveness of a wide

range of features, previously proposed for English. Similarly, a baseline approach in our work uses proven features in combination with a linear model and is shown to perform well across both Slovene datasets but still worse than the approaches using deep learning.

Lately, the approaches for languages other than English are also starting to shift towards the use of deep learning. Park et al. [20] use word embeddings and a feed-forward neural network to model coreference resolution as a binary classification problem and show its effectiveness for the Korean language, while Nitoń et al. [21] experiment with deep learning approaches that use a combination of word embeddings and handcrafted features and either a fully-connected neural network or a Siamese network [22] in a mention ranking or entity-mention approach. Training deep neural networks typically requires a large dataset to tune the weights stably. For some languages, annotated resources are either not available or very scarce, which is one of the reasons why authors experiment with learning cross-lingual coreference resolution. For example, Urbizu et al. [23] present a coreference resolution system for the Basque language, which they train on an English corpus. They compare the cross-lingual system with a monolingual (Basque) one and show that the cross-lingual system works slightly better. Similarly, although motivated by language similarity instead of data scarcity, Cruz et al. [24] present a coreference resolution system for Portuguese, which they learn on a Spanish corpus. They are able to achieve competitive performance to a monolingual system, trained on Portuguese.

Our work draws inspiration from existing literature and studies it in terms of the Slovene language. To the best of our knowledge, there currently exist no Slovene coreference resolution systems based on deep learning. In addition to this, our work is the first to analyze coreference resolution systems on the SentiCoref dataset [25].

3. Coreference Resolution Datasets

The majority of the state-of-the-art systems were evaluated on specialized shared tasks at MUC (Message Understanding Conference) [26], ACE (Automatic Content Extraction) [27], SemEval2010 (Semantic Evaluation) [28], and at CoNLL-2011 and CoNLL-2012 (Conference on Computational Language Learning) [29,30]. Nowadays, datasets presented at these shared tasks or conferences still represent the main coreference resolution benchmark datasets. Recently, some specific coreference resolution datasets were produced, such as gender-focused coreference resolution [31], commonsense-related coreference resolution [32] and coreference resolution as a part of general language understanding dataset [33].

In our experiments we use two Slovene coreference resolution datasets: coref149 [19], containing 149 documents, and SentiCoref [25], containing 837 documents. First, we provide some general statistics for both datasets and compare them to commonly used English datasets. Then, as our work presents the first analysis on SentiCoref, we provide a more detailed description of the dataset in Section 3.1.

We provide general statistics for both used datasets in Table 1. In addition, we note statistics for some other commonly used English datasets. We can see that coref149 is comparably small to the other datasets, being composed of less documents and containing less tokens. On the other hand, SentiCoref 1.0 dataset contains more documents than ACE 2004 and SemEval2010 which seems promising for training coreference resolution

models for Slovene. Most of the corpora (except coref149) are made up of news documents.

Table 1. Dataset statistics for the Slovene (coref149 and SentiCoref 1.0) and most often used English (ACE 2004, SemEval2010 and CoNLL-2012) coreference resolution datasets.

Statistic	coref149	SentiCoref 1.0	ACE 2004	SemEval2010	CoNLL-2012
Documents	149	837	450	314	2,135
Tokens	26,960	433,139	191,387	102,952	1,468,889
Entities	1,277	14,572	12,439	20,921	37,330
Trivial	831	7,721	-	-	-
Mentions	2,329	42,738	29,724	28,242	174,437
Overlapping	196	4,212	-	-	-

Interestingly, the ratio of tokens per document is similar among all datasets. The number of entities per document is comparable between SentiCoref 1.0 and CoNLL-2012, while it is lower for ACE 2004 and SemEval2010. Such rough comparison can provide an initial insight into whether SentiCoref 1.0 dataset is on par with the commonly used English datasets.

It is important to notice that there are a number of differences between the Slovene and English language. Apart from the fact that Slovene is a highly inflected language, it introduces verb as a new mention type. In Slovene texts, references to entities are often implicitly hidden in verbs and not mentioned explicitly as in English. Due to annotation specifics (which we describe in more detail in Section 3.1), we also report the number of trivial entities and overlapping mentions. Trivial entities contain only one mention in a document, while overlapping mentions are mentions that overlap in tokens, although they can refer to different entities. For example, the text "*Slovenian football club Olimpija*" contains three mentions ("*Slovenian*", "*Olimpija*" and "*Slovenian football club Olimpija*"), which refer to two entities (Slovenia and football club Olimpija).

3.1. SentiCoref 1.0 Dataset

In this section, we provide a more detailed description of SentiCoref 1.0, a dataset that was created to enable Slovene coreference resolution experiments on a larger scale. It is publicly available online [25].

For SentiCoref 1.0 we selected 837 articles from the existing SentiNews 1.0 corpus [34] which consists of 10,427 manually annotated Slovenian news articles for sentiment analysis. The content represents online news related to politics, business, economics and finance. The news were randomly sampled from Slovenian online news portals 24ur, Dnevnik, Finance, RTVSLO and Žurnal24. In SentiNews, each article is independently annotated by between two and six annotators for sentiment analysis using a five-level Lickert scale (very negative, negative, neutral, positive and very positive) on three levels of granularity (document, paragraph and sentence level). For SentiCoref, we selected documents from SentiNews that contain between 50 and 73 named entities, as detected

by Polyglot [35]. In Figure 1 we show a part of an annotated document from the dataset. It contains three types of annotations, which we describe next.

Named entity annotation: The basis for coreference resolution and target-level sentiment analysis are entities. In the corpus we therefore focused only on entities that contain at least one named entity mention in a document. This means that entities never explicitly mentioned in the corpus are not taken into account (e.g. if the entity is always referred to using pronouns). Based on the existing Slovene named entity recognition dataset [36] we decided to annotate:

- **(a) persons or groups of persons:** For example [Alfred Nobel], [poslanec SKD] (eng. parliament member from the SKD party) or [zamejci] (eng. Slovenes abroad).
- **(b) organizations:** For example [Švedska centralna banka] (eng. Swedish Central Bank). This category also includes political parties, for example [SKD] (SKD party).
- **(c) geographical names:** For example locations, such as [Maribor] and [Washington], political geographical units, such as [EU].

Coreference resolution annotation: Coreferences are annotated only for entities that contain at least one named entity mention in a document and represent identity-level coreferences. Thus, each coreference chain refers only to one specific underlying entity and not, e.g., a part-whole concept.

Target-level sentiment analysis annotation: One of the aims of the dataset was also to provide sentiment annotation for each entity in a document. As an entity is represented as a list of coreferent mentions, the task is to identify the sentiment of an entity in the context of a document. So, if there is a description of a crime that a person committed, then such entity would be annotated as a negative entity. Annotations for the entities are added to the last mention of an entity in a document.

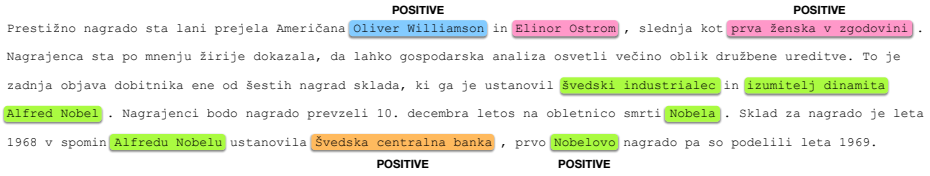

Fig. 1. Part of an annotated document from SentiCoref. Each entity and its coreferences are marked with the same color. Sentiment annotation is marked at the last mention of an entity. The English translation of the text is: "*The prestigious award went to Americans Oliver Williamson and Elinor Ostrom, the latter being the first woman in history to receive it. According to the jury, the winners have shown that economic analysis can shed light on most forms of social regulation. This is the latest announcement of the winner of one of six awards given by the organization, founded by the Swedish industrialist and inventor of dynamite Nobel. The winners will receive the prize on 10. December on the anniversary of Nobel's death. The Prize Fund was established in 1968 in memory of Alfred Nobel by the Swedish Central Bank, with the first Nobel Prize being awarded in 1969.*"

In Table 2 we show the general statistics of named entity types and sentiment values. Note that there is a difference of 451 entities between Table 2 and Table 1. This is the number of entities that do not have a sentiment value annotation. The lack of annotations was discovered after the end of the annotation campaign.

Table 2. Number of entities by their type and sentiment in the SentiCoref 1.0 dataset.

	Positive	**Neutral**	**Negative**	**All**
Person	637	2,611	542	3,790
Organization	756	4,455	986	6,197
Location	274	3,603	257	4,134
All	1,667	10,669	1,785	14,121

The dataset was annotated by a total of eight different annotators, with each document being annotated by two different annotators. All the documents were then manually curated by the second author of this paper. Compared to English datasets, SentiCoref 1.0 contains the following specifics.

- It contains annotations for overlapping mentions, the number of which we provide in Table 1. These can appear as mentions of different entities or the same. The latter are mostly left predicate complements (premodifiers), for example, "*[head of engineering [Zoran Arnež]*$_1$*]*$_1$" contains two overlapping mentions referring to the same entity. On the other hand, in case of right predicate complements (i.e. postmodifiers), there is always a character between the two mentions, such as '-', 'v', '(', '/' or ',. Such apposition is for example "*[Zoran Arnež]*$_1$*, [head of engineering]*$_1$."
- In Slovene, the mentions can implicitly be hidden inside a verb. In such cases, we annotate part of verbs that contain information about the entity. Such an example is the text "*[Postal je]*$_1$ *učitelj*", which would be directly translated into English as "*[Became]*$_1$ *a teacher*", although it is implied that the statement is about a man. These annotations exist only in cases where no explicit mention of an entity exists in a sentence. Another example is shown in Figure 2.

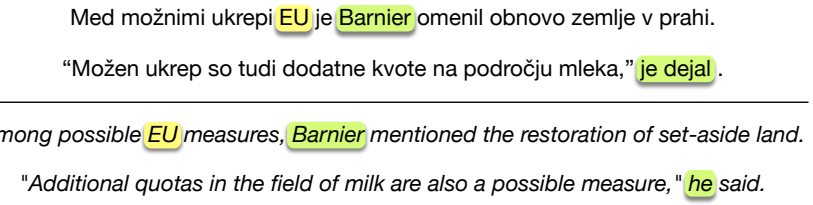

Fig. 2. An example of a Slovene coreference where a coreferent mention is "hidden" within a verb. The figure shows two entities and three mentions. The bottom part is the English translation of the Slovene example.

In Figure 3 we show the part of speech tag distribution in SentiCoref 1.0. We used Stanza [37] to annotate the corpus automatically. In the case of a multi-word mention, we take the type of the first word as the tag of the mention. We observe that nouns are most common, also because named entities are nouns. The next are adjectives which often play the role of a premodifier of a mention. The third and fourth are verbs and pronouns. Compared to English, verbs in Slovene implicitly contain pronoun mentions which are always explicit in English, so these would be represented as one group in English datasets. Other part of speech types are rare and represent special cases that appear at the beginning of mentions, for example, titles ("dr. Lahovnik") or abbreviations ("B. Bonnaud").

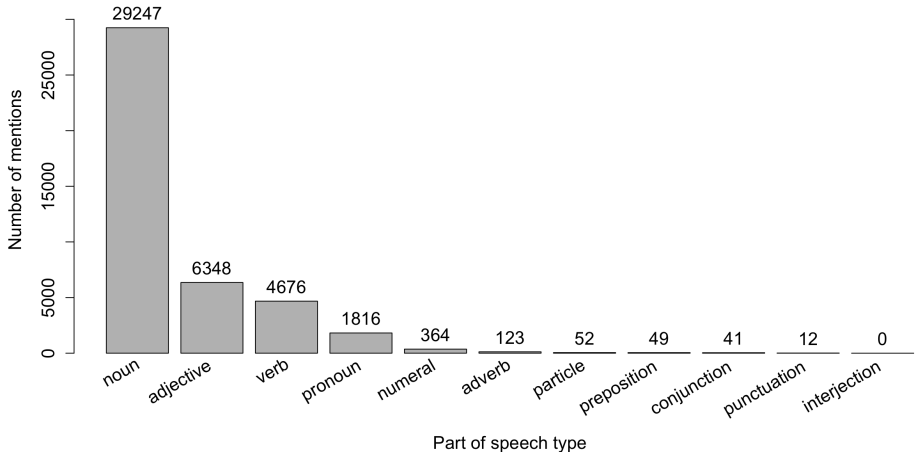

Fig. 3. Distributions of part of speech types of first words of the mentions in SentiCoref.

Lastly, we show three additional distributions for SentiCoref 1.0 in Figure 4: entity size, document size and distance between coreferent mentions. As described in Table 1, there are a lot of trivial entities in the dataset (around 50%). Still, entities containing up to 10 mentions are well represented and mostly contain other half of entities. The distribution of distances between two consecutive coreferent mentions (upper right) is important as it explains the maximum possible performance of a coreference resolution model that can take up to N consecutive mentions as input. For example, distance 0 means that mentions are directly consecutive (no other mentions in between), and distance 1 means that there is one other mention in between. We can observe that by collecting mentions up to a distance of 10 we could address most of the existing coreferences (around 95%).

As we selected only documents that contain at least 5 named entities, the minimum number of mentions per document is larger than that (i.e. 13 mentions). From Figure 4 we observe that most of the documents contain between 30 and 70 mentions. There exist documents with up to 145 mentions, but these are less frequent. These documents are typically sports game reports where a number of players and sports clubs are mentioned.

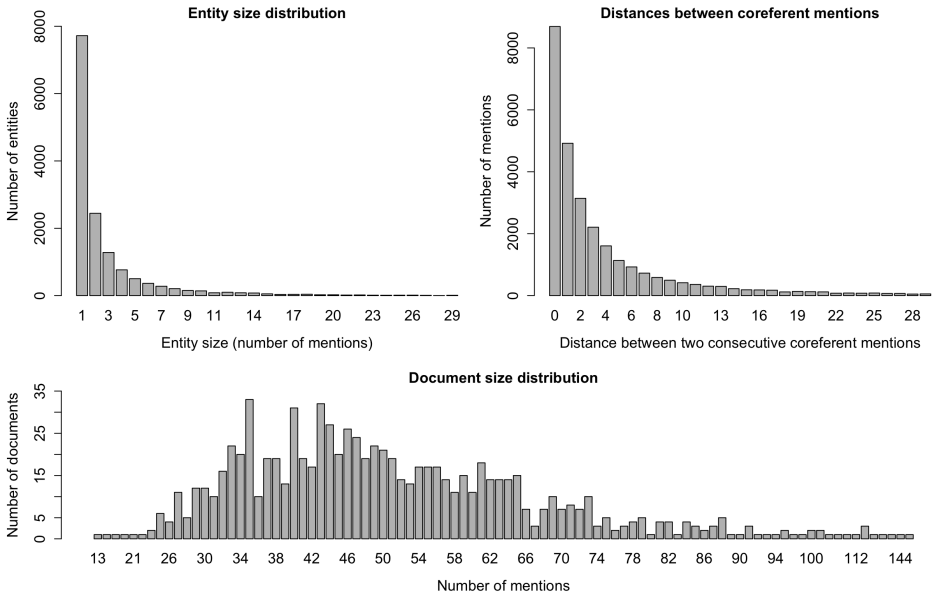

Fig. 4. Distributions of entity sizes in SentiCoref based on the number of mentions (upper left), distances between consecutive coreferent mentions (upper right) and document sizes based on the number of mentions (bottom).

4. Methods

4.1. 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 the 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 5 shows an example of a mention ranking algorithm.

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

4.2. Baseline Model

Our baseline model is a linear mention pair scorer based on handcrafted features. Scores are obtained for every antecedent candidate appearing in the document and then normalized using the softmax function. For constructing the features, we use additional metadata such as part of speech tags and lemmas. For coref149, this metadata is provided in the

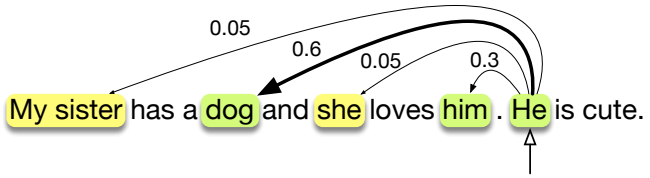

Fig. 5. Mention ranking algorithm. Marked words represent mentions of two different entities, split by color, based on the entity they reference. Mention currently being processed is "He". We compute scores for all antecedent mentions. The mention with the highest score is selected as a coreference.

ssj500k dataset, while for SentiCoref, this metadata is not provided, so we obtain it automatically using the Stanza library [37]. The features we use in our baseline model are based on already-proven ones reported in existing literature [5]. They are described in Table 3. Categorical features are encoded into binary ones using one-hot encoding. In the following sections, we refer to this approach as *linear baseline*.

Table 3. Features used in our linear 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
same number	one-hot encoded vector for values: match in number, don't match in number
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
jw dist	distance value between two mentions according to Jaro-Winkler metric

4.3. Neural Models

In this section, we first describe the used neural coreference scoring architecture. We describe it by detailing the process of obtaining the coreference score for a given mention and a coreference candidate. Next, we present our three variations of the architecture, which differ in the type of embeddings, used to represent the mention tokens.

Our neural architecture follows the neural network-based scorer, originally introduced as part of an end-to-end system for coreference resolution [9]. The scorer is shown schematically in Figure 6 and described next.

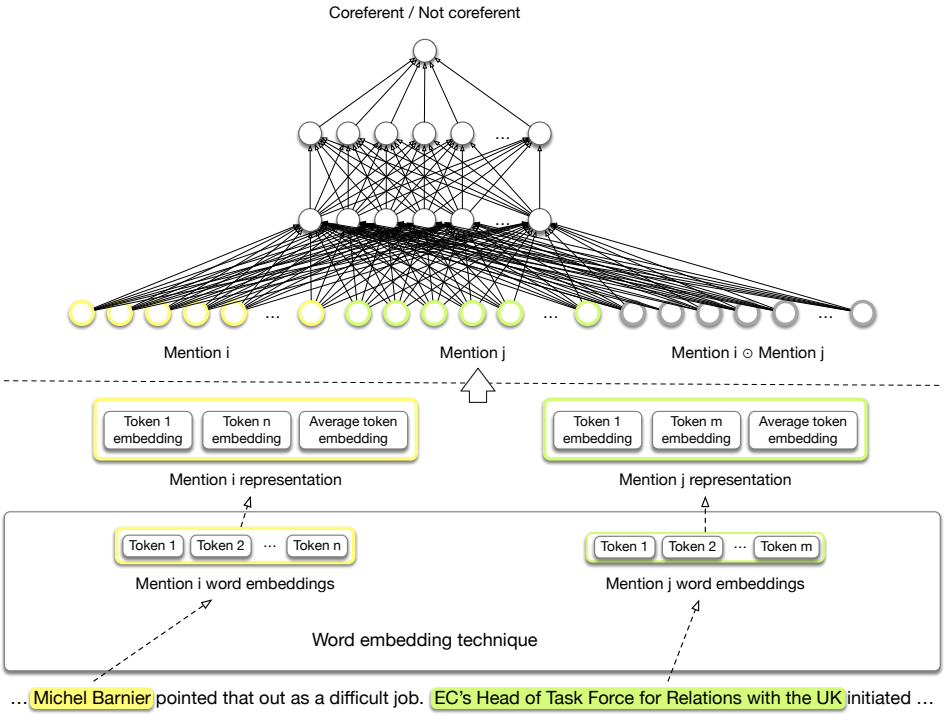

Fig. 6. Neural coreference scorer architecture. Input to the scorer represents both mention representations and their element-wise product. Mention representations consist of first mention token embedding, last mention token embedding and average token embedding of mention tokens.

The input to the scorer are tokens for a mention, and one of the candidate mentions for coreference, while the output is a coreference compatibility score between two mentions, representing how likely it is that the two mentions are coreferent. First, the tokens are embedded using one of the embedding types described later in this section. Then, a three-part mention representation is constructed independently for each mention. This is done by concatenating the embedding of the first token of a mention, the embedding of the last token of a mention and a learned weighted combination of embeddings for all mention tokens. The first and second parts of the representation are used to capture the left and right context of a mention, while the third part is used as an approximate representation of the head word inside a mention. Once the mention representations are obtained for both mentions, a three-part mention pair representation is constructed by concatenating the representations of the first mention, the second mention and their element-wise product. Finally, this is fed into a two hidden layer feedforward neural network with rectified linear

unit (ReLU) activation function to produce a coreference compatibility score, which is then used in the mention ranking framework, described in Section 4.1.

One aspect of the neural architecture, which is still vaguely described, are the embeddings used to represent the tokens. We experiment with different types of embeddings to produce three variations of the previously defined architecture. Specifically, we use non-contextual (word2vec and fastText), contextual ELMo (Embeddings from Language Models) and contextual BERT embeddings. The process of obtaining these embeddings is shown schematically in Figure 7 and described next.

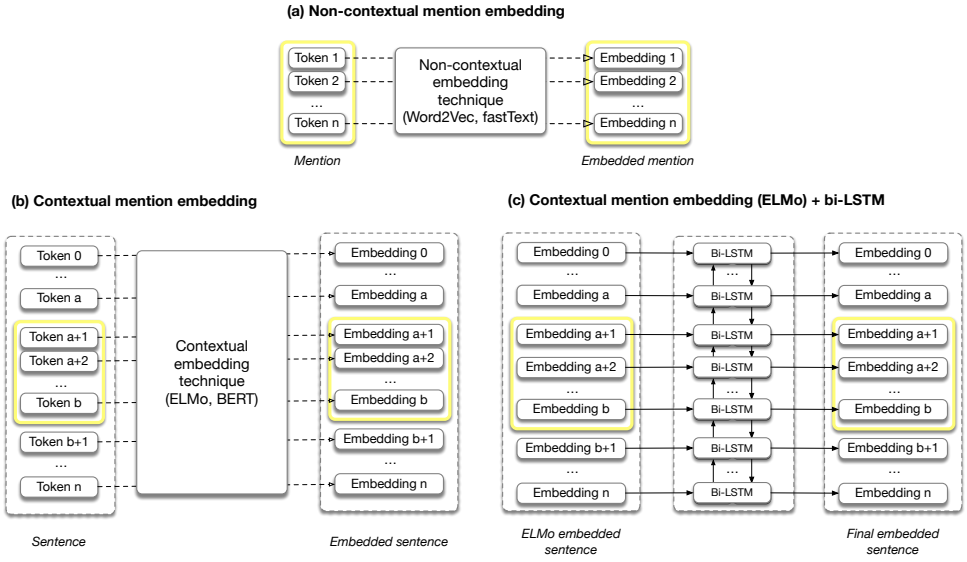

Fig. 7. Different embedding techniques used in our work. The input mention is marked in yellow. The figure shows methods to get (a) non-contextual word embeddings such as word2vec and fastText, (b) contextual word embeddings such as ELMo and BERT, and (c) the additional step that is used for processing ELMo embeddings: a pass through an additional bi-LSTM. For BERT-based embeddings (b), we use the output of its last hidden layer as embeddings.

Non-contextual Embeddings For experiments with non-contextual embeddings, we use word2vec embeddings [38] and fastText embeddings [39], which we provide in their original form as the input to the coreference scorer. Specifically, we use word2vec embeddings trained with the skip-gram architecture and fastText embeddings trained using the continuous bag of words architecture, a decision which we make based on the fact that such embeddings are already provided online. In our primary experiments, we use 100-dimensional word2vec embeddings [40] and 100-dimensional fastText embeddings, which we additionally fine-tune for coreference resolution. As we are dealing with datasets of very different sizes that might not both allow the learning of complex models,

we also experiment with a smaller (50) and bigger (300) dimensionality of word embeddings. For tokens appearing in the vocabulary that do not have an associated pretrained word embedding, we randomly initialize their embeddings to random $[0, 1)$ vectors of used dimensionality.

In the following sections we refer to these approaches as *word2vec* and *fastText*.

Contextual Embeddings: ELMo In the first approach using contextual embeddings, we use Embeddings from Language Models (ELMo) [41]. Following the setup used by the authors of ELMo, we learn a task-specific linear combination of the three ELMo layers. Additionally, we encode the resulting embedded document tokens using a bidirectional LSTM [42], processing each sentence independently. We use a pretrained Slovene ELMo model [43], whose weights we fine-tune together with the weights of coreference scoring module. In the following sections, we refer to this approach as *elmo-lstm*.

Contextual Embeddings: BERT In the second approach using contextual embeddings, we use BERT embeddings [13], following the setup described in existing literature [11], 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 segments of pre-determined maximum length and embed them independently. The embeddings we input to the coreference scorer correspond to the last hidden layer of BERT. 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. We use two types of BERT: a trilingual BERT model (CroSloEngual BERT) [44] and multilingual BERT. In the following sections, we refer to these approaches as *CroSloEngual BERT* and *multilingual BERT*.

5. Results and Discussion

In this section, we first explain the experimental settings and metrics used in our coreference resolution experiments. Then, we present the analysis of results obtained by our approaches on Slovene datasets.

5.1. Experimental Framework

There is no general agreement on which metric to use for the coreference resolution task. We adopt the most commonly used measures in the literature, which are described below. Prior to the measures we use in this paper, a graph-based scoring algorithm had been used that produced very unintuitive results [45,46]. There have been several metrics proposed, so we evaluate the system using the following most commonly used measures:

MUC The key idea in developing the MUC measure [47] was to give an intuitive explanation of the results for coreference resolution systems. It is a link-based metric (it focuses on pairs of mentions) and is the most widely used. MUC counts false positives by computing the minimum number of links that need to be added to connect

all the mentions referring to an entity. On the other hand, recall measures how many of the links must be removed so that no two mentions referring to different entities are connected in the graph. Thus, the MUC metric gives better scores to systems with more mentions per entity while ignoring entities with only one mention (singleton entities).

BCubed The BCubed metric [48] (B3) tries to address the shortcomings of MUC by focusing on mentions, and measures the overlap of the predicted and true clusters by computing the values of recall and precision for each mention. If k is the key entity and r the response entity containing the mention m, the recall for mention m is calculated as $\frac{|k \cap r|}{|k|}$, and the precision for the same mention, as $\frac{|k \cap r|}{|r|}$. This score has the advantage of measuring the impact of singleton entities, and gives more weight to the splitting or merging of larger entities.

CEAF The goal of the CEAF metric [49] is to achieve better interpretability. The result reflects the percentage of correctly recognized entities. We use entity-based metric (in contrast to a mention-based version) that tries to match the response entity with at most one key entity. For CEAF, the value of recall is $\frac{\text{total similarity}}{|k|}$, while precision is $\frac{\text{total similarity}}{|r|}$.

We report on precision, recall and F_1 score For each metric. Results are computed using *neleval*[2] package.

In addition, we also report on the **CoNLL 2012** score, which is the average F_1 score of the three metrics (i.e., MUC, B3 and CEAF) and is intended to serve as a compact summary of the model's performance. It was also used during CoNLL 2012 shared task [29] to rank participating coreference resolution systems. Unless noted otherwise, we use this metric to determine if method M_1 is better than method M_2.

We compute the described metrics using different evaluation techniques. On coref149, we use 10-fold cross-validation (CV), meaning we divide the dataset into 10 parts, train a model on 9 folds and evaluate it on the remaining fold. We repeat this 10 times, each time evaluating on a different fold, and report the mean score (along with the standard deviation) across the folds as the final result of a method. On SentiCoref, we instead decide to use a single split into a training, validation and test set in ratio 70%:15%:15%. We choose to do so primarily due to the substantially larger size of the dataset, which reduces the random fluctuation in the performance of the models. 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 loss on the validation set does not decrease for 5 consecutive epochs, the training is stopped, and the best state is used for evaluation. In each iteration of CV, an internal 3-fold CV is used in place of a validation set for hyperparameter and model selection.

5.2. Empirical Comparisons

The results achieved by presented methods are shown in Table 4 for coref149 and Table 5 for SentiCoref. 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

[2] Neleval package repository: `https://github.com/wikilinks/neleval` (Accessed on: April 9, 2021)

Table 4. MUC, B3 and CEAFe F1 scores of our approaches on the **coref149 dataset**, ordered by average F1 score. The numbers represent the means and standard deviations across 10 folds of CV.

Model	MUC	B3	CEAFe	Avg. F1
All-in-one	0.617 (0.070)	0.358 (0.046)	0.152 (0.029)	0.376 (0.047)
Each-in-own	0.000 (0.000)	0.688 (0.049)	0.562 (0.062)	0.417 (0.037)
fastText100	0.125 (0.090)	0.707 (0.041)	0.589 (0.050)	0.473 (0.043)
word2vec100	0.342 (0.099)	0.670 (0.100)	0.565 (0.113)	0.525 (0.048)
elmo-lstm	0.4246 (0.080)	0.7131 (0.038)	0.645 (0.042)	0.594 (0.035)
linear-baseline	0.539 (0.092)	0.793 (0.043)	0.701 (0.060)	0.678 (0.058)
multilingual BERT	0.719 (0.049)	0.841 (0.038)	0.801 (0.047)	0.787 (0.043)
CroSloEngual BERT	0.720 (0.081)	0.839 (0.033)	0.806 (0.031)	0.788 (0.039)

Table 5. Achieved MUC, B3 and CEAFe F1 scores of our approaches on the **SentiCoref dataset**.

Model	MUC	B3	CEAFe	Avg. F1
Each-in-own	0.000	0.525	0.389	0.305
All-in-one	0.770	0.231	0.050	0.350
linear-baseline	0.605	0.691	0.565	0.620
word2vec100	0.708	0.705	0.658	0.690
fastText100	0.778	0.773	0.753	0.768
elmo-lstm	0.855	0.819	0.810	0.828
multilingual BERT	0.923	0.891	0.886	0.900
CroSloEngual BERT	0.939	0.916	0.912	0.922

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 into a single cluster. Comparing these methods in isolation, we can see that the former has a higher average score on coref149, while the latter has a higher score on SentiCoref, which agrees with the statistics of trivial entities presented in Section 3: because coref149 contains a larger proportion of trivial entities, the "Each-in-own" model achieves a slightly higher score there.

Linear baseline achieves an average F1 score of 0.678 on coref149 and 0.620 on SentiCoref. It serves as a relatively strong baseline, beating both methods using non-contextual embeddings and one using contextual embeddings on coref149, where data is scarce. On SentiCoref, its performance is inferior to the mentioned methods since their weights can be more reliably tuned there. The results indicate that simpler methods based on manual feature engineering might be viable when we have a small amount of training data. Another desirable trait of the linear model is our ability to inspect what the model has learned by plotting the feature weights. The learned weights on Figure 8 indicate that string equivalence features (such as string match and suffix indicator) are universally useful, while the importance of some other features differs substantially.

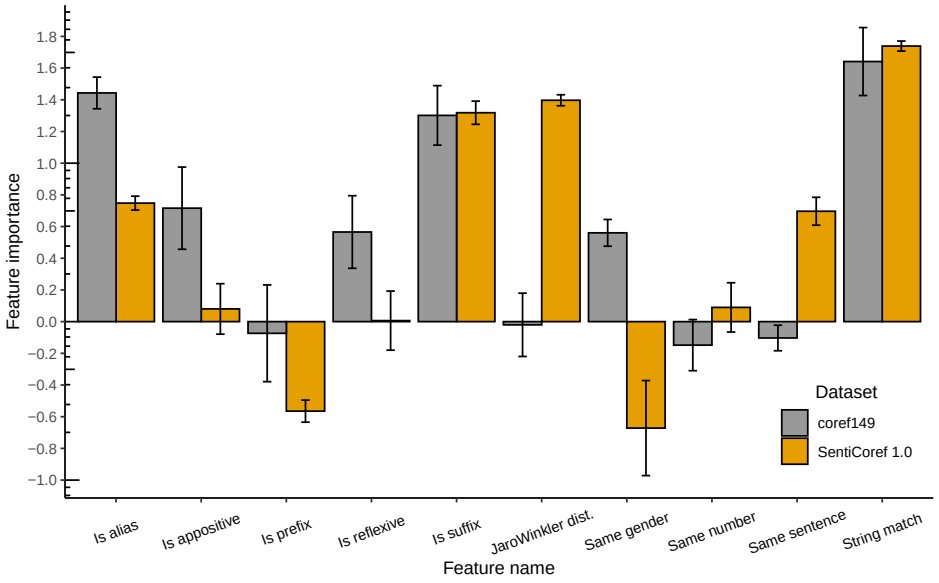

Fig. 8. The weights of linear baseline coreference scorer. For both datasets, the model assigns high importance to the string equivalence based attributes.

Although desirable on small datasets, the results achieved by linear baseline are surpassed by the neural approaches using either non-contextual or contextual embeddings once more data is available, as is the case with SentiCoref.

Focusing on the models using non-contextual embeddings first, we see that they achieve average F1 scores of 0.690 (word2vec) and 0.768 (fastText) on SentiCoref, while they achieve poor average F1 scores of 0.525 (word2vec) and 0.473 (fastText) on coref149. FastText embeddings offer increased flexibility in representing small variations of words due to being based on subword units, though they also amount to a larger amount of trainable weights than word2vec embeddings. The results seem to indicate that the fastText embeddings can not be tuned reliably on the coref149 dataset, so they perform worse than word2vec. Conversely, SentiCoref offers enough data to tune them, which results in a noticeable performance boost (+0.078 average F1 score over word2vec).

The outlined statement is further supported by our experiment with different non-contextual embedding sizes, the results of which are shown in Table 6 for coref149 and Table 7 for SentiCoref. On coref149, the top F1 scores are achieved by smaller embeddings (50-dimensional fastText and 100-dimensional word2vec) since they can be fit most reliably. On SentiCoref, approaches using fastText embeddings of all sizes outperform the approaches using word2vec embeddings. We note however that the two types of embeddings are obtained using different architectures, i.e. skip-gram and continuous bag of words. Some of the outlined differences could also be caused by this, though we do not explore the comparison further.

Table 6. MUC, B3 and CEAFe F1 scores of neural approaches using non-contextual embeddings of different dimensions on the **coref149 dataset**. The numbers represent the means and standard deviations across 10 folds of CV.

Model	MUC	B3	CEAFe	Avg. F1
fastText100	0.125 (0.090)	0.707 (0.041)	0.589 (0.050)	0.473 (0.043)
fastText300	0.132 (0.074)	0.706 (0.040)	0.591 (0.049)	0.477 (0.042)
word2vec50	0.361 (0.111)	0.607 (0.092)	0.479 (0.105)	0.483 (0.046)
word2vec300	0.210 (0.139)	0.680 (0.095)	0.568 (0.102)	0.486 (0.045)
fastText50	0.169 (0.090)	0.711 (0.041)	0.602 (0.059)	0.494 (0.059)
word2vec100	0.342 (0.099)	0.670 (0.100)	0.565 (0.113)	0.525 (0.048)

The results for models using contextual embeddings lead to similar conclusions. On coref149, the models using ELMo embeddings cannot learn many patterns and therefore achieve a poor average F1 score (0.594). This means that even the best score does not surpass the performance of a linear baseline on coref149. Surprisingly, the story is different for BERT models: the multilingual and trilingual BERT model approaches surpass the baseline and achieve a practically equivalent average F1 score (0.787 and 0.788).

Table 7. MUC, B3 and CEAFe F1 scores of neural approaches using non-contextual embeddings of different dimensions on the **SentiCoref dataset**.

Model	MUC	B3	CEAFe	Avg. F1
word2vec50	0.697	0.698	0.642	0.679
word2vec100	0.708	0.705	0.658	0.690
word2vec300	0.733	0.726	0.704	0.721
fastText50	0.768	0.765	0.748	0.761
fastText100	0.778	0.773	0.753	0.768
fastText300	0.785	0.788	0.759	0.777

On SentiCoref, the contextual models can be tuned much better, pushing the achieved scores far above those of non-contextual models and the linear baseline. The model using ELMo embeddings achieves an F1 score of 0.828, while the multilingual and trilingual BERT achieve F1 scores of 0.900 and 0.922, respectively.

The results show the overall effect of using different approaches to model coreference resolution, with the results for BERT on SentiCoref looking particularly impressive. To get additional perspective into the limits of our methods, we qualitatively observe the wrong predictions made by the best performing approach on SentiCoref and point out some error patterns which we observe multiple times, their likely causes and possible solutions. The examples which we refer to in descriptions of error patterns are also shown in Figures 9, 10 and 11.

– **Errors due to limitations of architectural decisions.** One type of error is due to the limited context made available to the BERT model. For example, the model assigns a mention at the end of a long document to a new entity, although the same entity was already detected at the start of the document. In our case, this is likely a consequence of representing the documents as independent segments of a fixed maximum size. The dilemma of how to represent long documents is still an open problem, although one possibility to reduce the number of such errors could be to represent documents as a combination of partially overlapping segments of maximum size, as outlined in work by Joshi et al. [11].
  The second type of error we mention here are the locally consistent but globally inconsistent assignments. For example, consider a document with the following three mentions (Figure 9): “Šrot” (in this case implying a man’s surname), “nadzornik v odvisnih družbah” (meaning “supervisor”) and “Tone Turnšek” (another man’s name and surname). The model first assigns “Šrot” as the antecedent of “nadzornik v odvisnih družbah”. Next, it assigns “nadzornik v odvisnih družbah” as the antecedent of “Tone Turnšek” . Although both of the assignments are potentially valid on their own, they form an inconsistency once both are taken together since the names clearly refer to different persons. The reason for these errors lies in the mention ranking framework, which does not explicitly consider existing entity assignments. Besides

changing the problem formulation, a possible improvement that tries to fix such inconsistencies is the use of an iterative refinement mechanism [10].

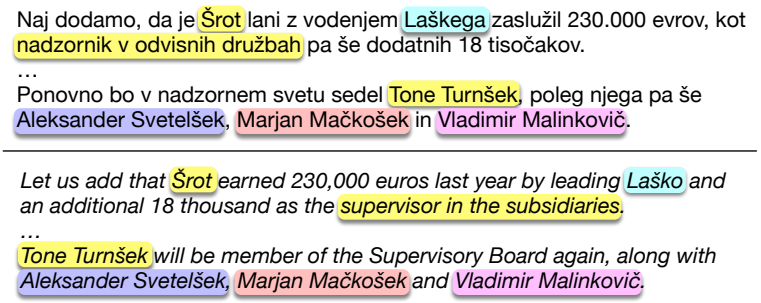

Fig. 9. Example of an error that the best BERT model makes on SentiCoref, likely due to limitations of architectural decisions. "Tone Turnšek" should have been assigned to a separate entity.

– **Lack of common sense.** For example (Figure 10), the model assigns the mentions "Merkur" (a Slovene company) and "nakelski trgovec" (meaning "a retail company based in Naklo") to a different cluster, although the two both refer to the same company. Such situations are arguably challenging even for humans if one does not have the background knowledge, and the modeling of common sense is still an open problem.

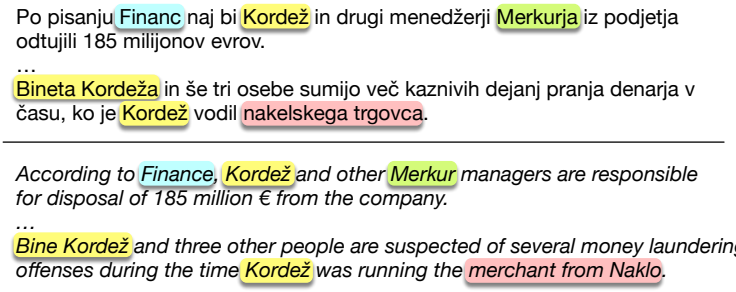

Fig. 10. Example of an error due to lack of common sense by the best BERT model on SentiCoref. "merchant from Naklo" should have been assigned the same cluster as "Merkurja."

– **Assignment of similar, but semantically different, named entities to same cluster.** For example (Figure 11), the mentions "Britanija" (meaning "Britain") and "Brioni" (a group of islands in Croatia) get clustered together, although they refer to two different geographical locations. This may be a consequence of the model putting too much emphasis on the common prefix "Bri" instead of taking into account the entire

words. A possible way to solve this could be to use a gazetteer to divide the mentions into two entities.

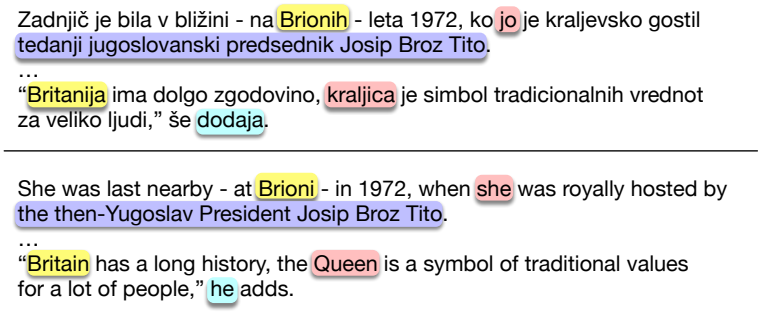

Fig. 11. Example of an error due to assignment of similar, but semantically different mentions by the best BERT model makes on SentiCoref. "Britanija" should have been a separate entity.

The quantitative results show that the highest scores obtained on the two datasets differ significantly. To see whether this gap can be narrowed, we perform additional experiments using augmented datasets. We expand the training subset of one dataset with all examples of the other dataset and rerun the training and evaluation procedure. Additionally, we perform cross-domain experiments, in which we take a model trained on one dataset and evaluate it on the other dataset without additional fine-tuning. The aim of this is to see how transferable the learned patterns are between datasets. We show the results in Table 8 for coref149 and Table 9 for SentiCoref and summarize them next.

The outcome can roughly be divided into two cases. The linear baseline performs equally or worse both with the augmented dataset as well as in cross-domain experiments. As seen in Figure 8, the weights for many features differ substantially between the datasets, so they cannot be set in a way that would benefit both datasets at once. The other approaches generally see a performance increase when using a dataset augmented with SentiCoref, and a comparable or worse performance when using a dataset augmented with coref149. The only model that benefits slightly from the augmentation with coref149 is the ELMo based model. Experimental results show that models trained on SentiCoref or an augmented dataset perform better on coref149 than those trained only on coref149, with the best trilingual BERT model achieving the new highest average F1 score (0.869). This strongly indicates that SentiCoref allows the models to learn more general patterns behind coreference. Therefore its use should be prioritized over coref149.

Throughout our experiments, the results show that once enough data is available, the methods using contextual embeddings (ELMo, BERT) start performing well and learn general patterns behind coreference. In our last set of experiments, we check the effect of certain architectural decisions on the performance of these methods on SentiCoref. Specifically, we observe the effect of three types of modifications:

Table 8. MUC, B3 and CEAFe F1 of our approaches in experiments involving augmented datasets (*augm.*) and cross-domain evaluation (marked as *SentiCoref* as the models are trained only on SentiCoref). The methods are evaluated on **coref149**, and the numbers represent the means and standard deviations across 10 folds of CV.

Model	MUC	B3	CEAFe	Avg. F1
linear-baseline (SentiCoref)	0.303 (0.077)	0.742 (0.044)	0.636 (0.058)	0.560 (0.047)
word2vec100 (SentiCoref)	0.468 (0.069)	0.675 (0.034)	0.589 (0.033)	0.578 (0.033)
word2vec100 (augm.)	0.506 (0.064)	0.713 (0.037)	0.625 (0.048)	0.615 (0.036)
linear-baseline (augm.)	0.491 (0.095)	0.781 (0.044)	0.683 (0.061)	0.652 (0.060)
fastText100 (SentiCoref)	0.539 (0.065)	0.790 (0.030)	0.728 (0.036)	0.686 (0.031)
fastText100 (augm.)	0.572 (0.101)	0.802 (0.042)	0.737 (0.054)	0.704 (0.060)
elmo-lstm (SentiCoref)	0.683 (0.063)	0.819 (0.037)	0.767 (0.042)	0.757 (0.040)
elmo-lstm (augm.)	0.705 (0.097)	0.850 (0.035)	0.816 (0.040)	0.790 (0.048)
multilingual BERT (SentiCoref)	0.787 (0.058)	0.856 (0.039)	0.826 (0.052)	0.823 (0.044)
multilingual BERT (augm.)	0.794 (0.050)	0.882 (0.039)	0.854 (0.050)	0.843 (0.031)
CroSloEngual BERT (augm.)	0.816 (0.073)	0.900 (0.028)	0.876 (0.039)	0.864 (0.043)
CroSloEngual BERT (SentiCoref)	0.826 (0.052)	0.904 (0.030)	0.877 (0.036)	0.869 (0.026)

- Does providing more context to the method using ELMo embeddings bring its performance closer to methods using BERT embeddings? To check this, we replace the independent encoding of sentences with the encoding procedure used in BERT-based models and instead encode non-overlapping segments of 256 words.
- How much does freezing the underlying embeddings and only fine-tuning the remaining layers decrease the performance?
- Does using a learned linear combination of all 12 hidden layers in BERT-based models improve the performance over using only the last hidden state?

The results of modified models are shown in Table 10. First, we can see that providing more context to the ELMo-based model has a negative effect, with its average F1 score decreasing by 0.016 in comparison to the model using a single sentence context. Besides decreasing the performance, the modification also increases the training time as the in-

Table 9. MUC, B3 and CEAFe F1 of our approaches in experiments involving augmented datasets (*augm.*) and cross-domain evaluation (marked as *coref149* as the models are trained only on coref149). The methods are evaluated on **SentiCoref**. Note that models marked with *coref149* are trained using a single 70%:15%:15% data split instead of CV in order to keep the scores comparable.

Model	MUC	B3	CEAFe	Avg. F1
fastText100 (coref149)	0.106	0.538	0.412	0.352
word2vec100 (coref149)	0.350	0.592	0.445	0.462
elmo-lstm (coref149)	0.547	0.546	0.512	0.535
linear-baseline (coref149)	0.611	0.694	0.564	0.623
linear-baseline (augm.)	0.608	0.693	0.568	0.623
word2vec100 (augm.)	0.668	0.703	0.647	0.673
multilingual BERT (coref149)	0.761	0.704	0.666	0.710
CroSloEngual BERT (coref149)	0.764	0.746	0.718	0.743
fastText100 (augm.)	0.783	0.776	0.755	0.771
elmo-lstm (augm.)	0.864	0.830	0.827	0.840
multilingual BERT (augm.)	0.911	0.890	0.885	0.895
CroSloEngual BERT (augm.)	0.921	0.890	0.881	0.897

creased number of words inside a segment means more words are processed sequentially using a LSTM. Second, freezing the underlying embeddings has a noticeable effect on BERT-based models and a small effect on ELMo-based models. The latter is a consequence of the model having an additional LSTM context encoder, which manages to act as a rough replacement for the trainable weights of ELMo. All three variations with frozen embeddings however still outperform the models using non-contextual embeddings. Last, we find that using a learned linear combination of all 12 BERT hidden layers instead of the last hidden layer does not improve the performance further. Multilingual BERT achieves a practically equivalent F1 score of 0.900, while the modified trilingual BERT sees a slight performance decrease with an average F1 score of 0.910.

6. Conclusion

We have introduced a new coreference resolution dataset for the Slovene language and performed experiments on it using variously complex models, showing that it allows us to learn strong models, to the point that they show strong performance even on a different dataset (coref149). Simultaneously, we have evaluated the methods on the existing

Table 10. MUC, B3 and CEAFe F1 scores of our approaches using contextual embeddings with three types of modifications: using 256-word context instead of a single sentence, using a learned linear combination of 12 hidden BERT layers and freezing of underlying embeddings (❄). The methods are evaluated on SentiCoref. For reference, we repeat the results of unmodified approaches (at the beginning).

Model	MUC	B3	CEAFe	Avg. F1
(elmo-lstm)	0.855	0.819	0.810	0.828
(multilingual BERT)	0.923	0.891	0.886	0.900
(CroSloEngual BERT)	0.939	0.916	0.912	0.922
elmo-lstm (segments of 256 words)	0.853	0.802	0.780	0.812
multilingual BERT ❄	0.828	0.799	0.801	0.810
elmo-lstm ❄	0.852	0.806	0.799	0.819
CroSloEngual BERT ❄	0.847	0.813	0.815	0.825
multilingual BERT (12 layers)	0.915	0.896	0.891	0.901
CroSloEngual BERT (12 layers)	0.934	0.903	0.895	0.910

(smaller) dataset and shown that its small size can present a problem for learning more complex models.

Although the best of the analyzed methods show surprisingly good results, it should be noted that we have only tackled the mention clustering part of the problem. The mention detection step undoubtedly introduces some noise to the process. As an example, the authors of the first end-to-end neural coreference resolution system [9] note that the average F1 score of their system increased by 0.175 when they replaced mention detection with oracle mentions. One of the logical next steps would be to check how well an end-to-end approach would work on Slovene data.

Additionally, knowing that the SentiCoref dataset is suitable for learning complex models, a possible next step would be to check if we could use it to aid the learning of coreference resolution for a different language that is similar to Slovene, for example Croatian.

Acknowledgments. The work presented in this paper started as a project in a natural language processing course and was then improved upon with additional experiments. We would like to thank Blažka Blatnik and Martin Čebular for their contributions to this project throughout the course. We would also like to thank the anonymous reviewers for the detailed reviews and helpful comments.

The SentiCoref 1.0 corpus preparation was funded by CLARIN.SI 2019 projects - *Corpus for Slovene coreference resolution and aspect-based sentiment analysis–SentiCoref 1.0.*

References

1. Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235, 2003.
2. Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.
3. Vincent Ng and Claire Cardie. 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, July 2002. Association for Computational Linguistics.
4. Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.
5. Slavko Žitnik, Lovro Šubelj, and Marko Bajec. SkipCor: Skip-mention coreference resolution using linear-chain conditional random fields. *PloS one*, 9(6):e100101, 2014.
6. Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, 2010.
7. Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 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, July 2015. Association for Computational Linguistics.
8. Xiaofeng Yang, Jian Su, GuoDong Zhou, and Chew Lim Tan. 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, aug 23–aug 27 2004. COLING.
9. Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
10. Kenton Lee, Luheng He, and Luke Zettlemoyer. 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)*, pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
11. Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 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, November 2019. Association for Computational Linguistics.
12. Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.
13. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics.

14. Maciej Ogrodniczuk and Mateusz Kopeć. Rule-based coreference resolution module for Polish. In *Proceedings of the 8th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2011)*, volume 191, page 200, 2011.
15. Voldemaras Žitkus, Rita Butkienė, Rimantas Butleris, Rytis Maskeliunas, Robertas Damasevicius, and Marcin Woźniak. Minimalistic approach to coreference resolution in Lithuanian medical records. *Computational and Mathematical Methods in Medicine*, 2019:1–14, 03 2019.
16. Svetlana Toldova, Ilya Azerkovich, Alina Ladygina, Anna Roitberg, and Maria Vasilyeva. Error analysis for anaphora resolution in Russian: new challenging issues for anaphora resolution task in a morphologically rich language. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 74–83, San Diego, California, June 2016. Association for Computational Linguistics.
17. Mateusz Kopeć and Maciej Ogrodniczuk. Creating a coreference resolution system for Polish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 192–195, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).
18. Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. BART: A modular toolkit for coreference resolution. In *Proceedings of the ACL-08: HLT Demo Session*, pages 9–12, Columbus, Ohio, June 2008. Association for Computational Linguistics.
19. Slavko Žitnik and Marko Bajec. Coreference resolution for Slovene on annotated data from coref149. *Slovenščina 2.0: empirical, applied and interdisciplinary research*, 6(1):37–67, Jun. 2018.
20. Cheoneum Park, Kyoung-Ho Choi, Changki Lee, and Soojong Lim. Korean coreference resolution with guided mention pair model using deep learning. *ETRI Journal*, 38(6):1207–1217, 2016.
21. Bartłomiej Nitoń, Paweł Morawiecki, and Maciej Ogrodniczuk. Deep neural networks for coreference resolution for Polish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).
22. Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.
23. Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 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, June 2019. Association for Computational Linguistics.
24. A. F. Cruz, G. Rocha, and H. L. Cardoso. Exploring Spanish corpora for Portuguese coreference resolution. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 290–295, 2018.
25. Slavko Žitnik. Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0, 2019. Slovenian language resource repository CLARIN.SI.
26. Lynette Hirschman and Nancy A. Chincor. MUC-7 coreference task definition. In *Proceedings of the Seventh Message Understanding Conference*, pages 1–17, San Francisco, 1997. Morgan Kaufmann.
27. George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program–tasks, data, and evaluation. In *Proceedings of LREC*, volume 4, pages 837—-840, 2004.
28. Marta Recasens, Lluís Màrquez, Emili Sapena, Antònia M. Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8, 2010.

29. Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 1–40. Association for Computational Linguistics, 2012.
30. Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1—-27, 2011.
31. Sandeep Attree. Gendered ambiguous pronouns shared task: Boosting model confidence by evidence pooling. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 134–146, Florence, Italy, August 2019. Association for Computational Linguistics.
32. Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012.
33. Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc., 2019.
34. Jože Bučar. Manually sentiment annotated Slovenian news corpus SentiNews 1.0, 2017. Slovenian language resource repository CLARIN.SI.
35. Rami Al-Rfou. Polyglot, April 2020.
36. Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. Training corpus ssj500k 2.2, 2019. Slovenian language resource repository CLARIN.SI.
37. Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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*, 2020.
38. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
39. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
40. Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 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, 2017.
41. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
42. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
43. Matej Ulčar and Marko Robnik-Šikonja. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4731–4738, May 2020.
44. Matej Ulčar and Marko Robnik-Šikonja. FinEst BERT and CroSloEngual BERT. In *Text, Speech, and Dialogue*, pages 104–111, 2020.

45. Nancy Chincor. MUC-3 evaluation metrics. In *Proceedings of the 3rd conference on Message understanding*, pages 17–24, Pennsylvania, 1991. Association for Computational Linguistics.
46. Nancy Chinchor and Beth Sundheim. MUC-5 evaluation metrics. In *Proceedings of the 5th conference on Message understanding*, page 69–78, Pennsylvania, 1993. Association for Computational Linguistics.
47. Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics, 1995.
48. Amit Bagga and Breck Baldwin. 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, 1998.
49. Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 135–es, USA, 2004. Association for Computational Linguistics.

Matej Klemen is a junior researcher at the University of Ljubljana, Faculty for computer and information science, where he is also a PhD student. He is mainly interested in the development of deep learning approaches for natural language processing. Additionally, he is interested in making the state of the art approaches applicable to languages that are not in the research spotlight, such as Slovene.

Slavko Žitnik is an assistant professor at the University of Ljubljana, Faculty for computer and information science. He is teaching courses related to databases, natural language processing, information retrieval and information systems. He is actively engaged in multiple research projects related to semantic technologies, such as coreference resolution or knowledge base constructions. He is cooperating with a number of research institutions - University of Belgrade, Sorbonné Universite Paris 1, University of South Florida and Harvard University. Apart from research he tries to productivize research results together with the needs of companies and this paper is a research result of such collaborations.

Received: November 20, 2020; Accepted: October 30, 2021.