SkipCor: Skip-Mention Coreference Resolution using Linear-Chain Conditional Random Fields

Slavko Žitnik[1,*], Lovro Šubelj[2], Marko Bajec[3]

1 Slavko Žitnik, Laboratory for Data Technologies, University of Ljubljana and Optilab d.o.o., Ljubljana, Slovenia

2 Lovro Šubelj, Laboratory for Data Technologies, University of Ljubljana, Ljubljana, Slovenia

3 Marko Bajec, Laboratory for Data Technologies, University of Ljubljana, Ljubljana, Slovenia

∗ **E-mail: Corresponding slavko.zitnik@fri.uni-lj.si**

Abstract

Coreference resolution detects groups of mentions in textual data that refer to the same real-world entities. It is one of the key information extraction tasks, while recent work shows that it can also support other tasks. Thus, we propose a novel coreference resolution system denoted SkipCor that reformulates the problem as a sequence labeling task. None of the existing supervised or unsupervised and pairwise or sequence-based models are similar to this proposed supervised sequence-based solution. This system is based merely on linear-chain conditional random fields, to enable high scalability with fast model training and inference, and a straightforward parallelization. Distant coreferent mentions in the text are detected using a special transformation of the data into skip-mention sequences with an intuitive O/C labeling scheme. We evaluate the proposed system against the ACE 2004, CoNLL 2012 and SemEval 2010 benchmark datasets and their subdatasets, where it clearly outperforms two baseline systems that detect coreferentiality using the same features. The first employs only a single mention sequence without further transformations, while the second simulates a traditional pairwise system. Furthermore, the obtained results are at least comparable to the current state-of-the-art in coreference resolution. Lastly, the paper also investigates the expected drop in accuracy in real-world scenarios, which has not been reported in the literature before.

Introduction

Information extraction (IE) gained importance in the 1970s, when early systems were focused mostly on the automatic detection of named entities in textual data [1]. Since then, a large number of IE systems dealing with entity extraction, relation extraction, and/or coreference resolution tasks have been proposed in the literature [2], along with the latest based on ontologies [3]. The coreference resolution task [4] is the task of detecting phrases in the text that refer to the same underlying entity [5] (i.e., the subject of the discussion that is then digressed, changed, etc.). These phrases are called *mentions*, and can be mentions of either named (e.g., John Doe), nominal (e.g., the guy with the glasses), or pronominal type (e.g., he or him) [6]. The goal of coreference resolution is thus to detect groups of mentions that refer to the same real-world entities. To accomplish this, one employs, apart from an initial text pre-processing, mention detection (i.e., identification of phrases that represent valid entity mentions), and mention clustering (i.e., determining which pairs of mentions corefer). Since the former can be solved in a rather straightforward fashion [7], we here consider only the last (we assume that the mentions in the text are given).

Early work on coreference resolution started in the late 1990s by employing decision trees and hand-written rules [8]. The majority of supervised approaches attempted to classify whether two arbitrary mentions within some document are coreferent. As most of the mention pairs are, obviously, not coreferent, the approaches learn from greatly unbalanced datasets, which can to some extent be addressed by different pruning techniques [9, 10]. Furthermore, researchers have also proposed different unsupervised

approaches [11], and the performance of the recent state-of-the-art unsupervised systems is comparable to supervised ones [12, 13]. It should, however, be stressed that the most important part of any coreference resolution system is the set of feature functions adopted [10, 13, 14] (i.e., functions that generate features which are additional attributes of the input data and are then used to better model the coreference resolution task), which can greatly affect its performance.

Coreference resolution is considered one of the key IE tasks [15] since it provides long-distance contextual information (i.e., mutual information between mentions that can be separated by many words in the text). In fact, various information extraction tasks (e.g., entity or relation extraction) can gain from an automatic identification of related parts of the text [16]. For instance, while one part can be associated with a relation, the other can be enriched with an attribute; therefore, named entity recognition can be improved using the contextual information from both parts. Identification of coreferent mentions in the text has thus already proved useful in various domains, ranging from mining news articles [17] to biological data [18].

In this paper, we propose a skip-mention coreference resolution system referred to as 'SkipCor'. The system is based merely on the linear-chain conditional random fields algorithm [19], whereas distant coreferent mentions in the text are detected using a novel transformation of the data into skip-mention sequences. In particular, an n skip-mention sequence is defined as a sequence of every $n+1$th mention in some document. These allow the use of very simple first-order (i.e., linear-chain) models that enable much faster and exact training and inference than do the general models. Thus, in contrast to most other approaches, the proposed system is completely parallelizable with a linear time complexity (in the number of mentions in the text). We compare SkipCor to a baseline system, on seven standard benchmark datasets. It clearly outperforms the baseline system that uses only a single sequence of mentions and a standard pairwise system that, as in traditional approaches mentioned above, looks at all the mention pairs in order to identify the coreferent ones. Furthermore, the results obtained are at least comparable to the current state-of-the-art in coreference resolution. We also investigate the drop in accuracy to be expected in real-world scenarios, where systems are trained on one dataset, and adopted on another, something which may be of independent interest.

Background

The majority of techniques for coreference resolution transform the problem into a pairwise classification task [9, 10] (i.e., the algorithm checks every pair of mentions for coreference). This enables the use of standard machine learning classifiers that rely on hand-labeled data sets. On the other hand, unsupervised techniques infer the coreferentiality based on sequences of mentions [7, 13], which are much harder to train and are not easily generalized to new problems or domains. In this section we will provide an overview of the different coreference resolution systems, with special focus on approaches based on graphical models [19] (as SkipCor).

One of the earliest supervised approaches used a decision tree algorithm and twelve informative feature functions [20]. That approach was the first to improve on the performance of previously state-of-the-art rule-based techniques. Even though the adopted features were based solely on pairs of mentions with local information, it was difficult to improve their results by only using more sophisticated algorithms. Therefore, a number of innovative and linguistic-rich feature functions [10, 14] along with different algorithms like maximum entropy [21], SVM classifiers [22] and Markov Logic Networks [23] have been proposed in the recent literature. Recently, Bengston and Roth [14] have systematically divided different feature functions into categories and clearly demonstrated their importance. In particular, they have shown that the development of well-designed features can greatly improve the performance of a coreference resolution system. Due to the similarities among the proposed supervised systems, the Reconcile platform [24] was developed in order to provide a common framework for new algorithms, features, and their evaluation.

Unsupervised approaches demand no training data. Nevertheless, state-of-the-art systems still achieve

comparable results to the supervised systems. Haghighi and Klein [13] proposed a modular unsupervised system using rich features. The system is based on a three-step procedure, consisting of the extraction of syntactic paths from the mentions, the evaluation of semantic compatibility between the mentions, and the selection of reference mentions, which serve as the basis for using pairwise decisions over transitive closures. Lee et al. [7] upgraded Raghunathan's system [12], which is based on a multi-pass sieve approach. They employed thirteen sieves (i.e., sequential processing steps) sorted by precision. During the execution of each sieve, the entire dataset is processed by applying a few manually written patterns. These hand-crafted patterns relate only to syntactic parse trees and extracted named entities, and are based on different heuristics and dataset specifications. Some unsupervised techniques have also been proposed. They infer coreferentiality based on sequences of mentions [25–27].

In the field of factor graphs, McCallum et al. [28] proposed three general conditional random fields (CRF) models to solve the coreference resolution problem. The first is a general model (i.e., the CRF structure is unrestricted) and the training or inference is therefore complex. In such cases exact inference is not possible and therefore approximation algorithms must be used to compute right marginal values for the underlying CRF structure [29]. The second model represents pairs of mentions by specific attributes, while the third represents the pairs as nodes in the model. Wellner et al. [30] successfully applied coreference resolution to citation matching, interestingly by using a special case of McCallum's first model combined with named entity extraction. Most similar to the linear models, a skip-chain CRF has been proposed in [31], which also supports the use of long-distance dependencies by incorporating additional cliques into the model. Still, longer times are needed for training and inference compared to linear-chain CRF. Cullota et al. [9] proposed the use of first-order probabilistic models over sets of mentions; thus, the algorithm operates directly on the entities. To avoid a combinatorial explosion of all possible entity subsets, they incrementally merged different mentions into sets. Later, they also included the step of canonicalization [30], which refers to the process of generating the underlying entities along with their attributes. Recently, Sundar et al. [32] proposed a CRF-based coreference resolution system. They further decomposed the problem into two subtasks: pronominal resolution using general CRFs that has only parse tree features, and non-pronominal resolution using linear-chain CRFs that has different string similarity features. Although the system is based on linear models, the input to the models still consists merely of sequences of length two.

In Table 1 we show the classification of some of the coreference resolution approaches that have been put forth in the literature. We categorize the systems along two dimensions: the type of input to the algorithm, and the type of model learning. As can be observed, the proposed SkipCor system is novel from the perspective of the selected dimensions. Among the unsupervised approaches, coreference resolution systems have been developed for both pairwise and sequence-based input types. In contrast, supervised approaches have mainly employed only pairwise comparisons. The system in [32] is similar to our baseline algorithm, SkipCorPair; however, it predicts whether two mentions are coreferent using a CRF algorithm. Also, [28] presents some CRF-based methods, but it evaluates only a version where each node represents a pair of mentions.

In summary, SkipCor represents a novel CRF-based approach that identifies coreferences over mention chains and employs simple clustering to uncover all mentions in the text that refer to the same entity. In contrast to other systems, we adopt a supervised algorithm for training and inference on sequence-based data. Thus, instead of using a pairwise or set-based approach, we consider sequences of mentions in some document and use simple linear-chain CRF models. To enable the use of such simple models, we introduce an adequate transformation of the data into skip-mention sequences. Consequently, the feature functions also refer to non-local information and can detect distant mention coreferences. Note also that the training and inference of linear-chain CRFs can be solved with a fast and exact algorithm, which significantly reduces the time complexity of the system.

Conditional Random Fields

Conditional random fields (CRF) [19] is a discriminative model that estimates the joint distribution $p(\overline{y}|\overline{x}, w)$ over the target sequence $\overline{y}$ conditioned on the observed sequence $\overline{x}$ and weight vector w (see below). We represent a sentence by a sequence of words x_i with additional corresponding sequences that represent attribute values such as part-of-speech tags $x_i^{k_1}$, lemmas $x_i^{k_2}$, relations $x_i^{k_3}$, and other observable values $x_i^{k_j}$. These values are used by feature functions f_l that are weighted during CRF training in order to model the target sequence $\overline{y}$. The sequence $\overline{y}$ corresponds to the source sequence and consists of the labels that we would like to automatically infer. For named entity recognition, we commonly use tags such as PER for person type, ORG for organization, and LOC for location. Similarly for relationship extraction, we use tags WORKS-AT, LIVES-IN, etc. For the coreference resolution task, we build sequences containing only mentions, as opposed to sequences containing all the words in a document. Then we use the label C if the current mention is coreferent with the previous one, and O otherwise.

In the field of IE, CRFs have been successfully employed for various sequence labeling tasks and have achieved state-of-the-art results. It can also deal with a large number of multiple, overlapping, and non-independent features.

Training a CRF is thus maximizing the conditional log-likelihood of the training data, by which we find a weight vector w that predicts the most probable sequence $\hat{y}$ for a given $\overline{x}$. Hence,

$$\hat{y} = \arg\max_{\overline{y}} p(\overline{y}|\overline{x}, w) \tag{1}$$

where the conditional distribution is

$$p(\overline{y}|\overline{x}, w) = \frac{\exp\left[\sum_{l=1}^{m} w_l \sum_{i=1}^{length(\overline{x})} f_l(\overline{y}, \overline{x}, i)\right]}{C(\overline{x}, w)} \tag{2}$$

Here, m is the number of feature functions and $C(\overline{x}, w)$ is a normalization constant computed over all possible sequences $\overline{y}$.

The structure of a CRF defines how the dependencies with target labels are modeled. A general graphical model (i.e., a graph denoting the conditional dependence structure) can depend on many labels and is therefore intractable for training or inference without complex approximation algorithms. Thus, we use only a simple linear-chain CRF (LCRF) model, which depends on the current and previous labels (i.e., a first order model). The structure of such a model is represented in Figure 1. Furthermore, with the use of a number of feature functions and special dataset transformations, our method achieves comparable results to the best known systems.

Methods

In this section we introduce the proposed SkipCor algorithm. First, we overview and introduce new feature functions used by conditional random fields models in the present paper. Next, we explain the data representation using skip-mention sequences and illustrate the coreference resolution execution of the proposed system on an example document. We also support the proposed representation by examining the distribution of consecutive coreferent mention distances on a representative coreference dataset. Last, we explain the implementation [33] of the proposed SkipCor system and discuss the time complexity of the algorithm.

Feature Functions

The selection of informative features is the main source of an increase of precision and recall when training machine learning classifiers. Feature functions are usually implemented as templates and the final features are then generated by scanning the entire training data. In natural language processing, a few thousand or more features are commonly used, which can be efficiently handled by a CRF. A feature function that returns 1 if the current mention is of person type or the previous mention is equal to "Mr." and 0 otherwise, is defined by:

$f_l(\overline{y}, \overline{x}, i)$ `= if (`$y_i ==$ PER $\vee\, x_{i-1} ==$ "Mr.")` `then return 1 else return 0`

Although many feature functions have been proposed in the literature [5, 14, 20, 34–36], we introduce new feature functions for the purpose of this research. These can be sorted into the following categories:

Preprocessing These feature functions use standard preprocessing labels, which are a result of the preprocessing step, such as lemmas, part-of-speech (POS) tags, chunks, and parse trees. The derived feature functions are "target label distribution", "do POS tags match on distances up to two mentions away", "distribution of POS tags", "mention type match", "is a mention pronoun of demonstrative/definitive noun phrase", "is mention a pronoun", "length between mentions within a parse tree", "parse tree path from the root node", "parse tree path between the two mentions", "depth of a mention within a parse tree", and "parse tree parent value match".

Location Sometimes it is important to know where the mention resides. Location feature functions deal with the mention's location compared to the whole document, sentence, or other mentions. Our approach already implicitly uses mention distance at each skip-mention model, but we still employ some specific feature functions. These are "sentence/mention/token distance between the two mentions", "is first/last mention" and "are mentions within the same sentence".

Mention Shape Mention constituents are represented as word phrases and by using mention shape features we are interested in whether two of them share some property. These feature functions are string-based and are implemented as follows: "does a mention start with an upper case", "do both mentions start with upper case", "does a prefix/postfix/whole of left/right mention on distances up to five mentions match", "does a mention text/extent match", "is one mention appositive of another", "is one mention prefix/suffix/substring of another", "Hearst mention co-occurrence rules", "is a mention within quotes", "does a mention contain head/extent words of another" and "length difference between the two mentions".

Semantic This class of feature functions captures semantic relationships between mentions by employing additional semantic sources, such as WordNet [37], specialized lexicons, semantic gazeteer lists, and ontologies. The semantic feature functions are "do named entity types match", "do mentions agree on gender/number" [38], "is one mention appositive of another", "is a mention an alias of another" (heuristically), "edit distance similarity between two mentions", "WordNet relation (hypernym/hyponym/synonym) between the mentions", "do mentions share the same WordNet synset", "current mention word sense", "do both mentions represent an animate object" [39] and "do both mentions speak" (taking context words into account).

A brief description and exact list of feature functions that we use is presented in Table 9. Still, their exact implementations can be retrieved from our public source repository [33] (within the class `FeatureFunctionPackages`).

Skip-mention Sequences

Since merely linear-chain CRF models are used, we can identify only coreferences over two directly consecutive mentions. Thus, to detect coreferences over mentions on larger distances, i.e., having one, two, three, or more mentions in between, we propose a skip-mention dataset transformation.

To support our transformation idea, we show the distribution of distances between two consecutive coreferent mentions (see Figure 2) in the SemEval2010 evaluation dataset. Although the figure shows the distribution for only one dataset, it is representative enough to illustrate the general problem, which is the same for all other datasets. According to the distribution, only 10% of the directly consecutive mention pairs are coreferent, while the highest number (i.e., 12.5%) of coreferent mention pairs are at distance one - i.e., having one other mention in between. Taking into account all mention pairs up to a distance of 20, cumulatively, 81% of the mention pairs can be identified. With distances up to 50, about 92% of the mention pairs can be identified. However, by using longer or all possible distances, the accuracy of a general coreference system is not expected to increase since more false positives are extracted. To overcome such problems, a promising cut-off point is selected (see Figure 7).

Thus, to detect coreferences we form a zero skip-mention sequence from each document, which contains all the mentions from a document. Then we form specific s skip-mention sequences. Each s skip-mention sequence contains every $(s+1)$-th mention from a document and one linear-chain CRF model is trained for each value of s. In the next section we present an example of detecting coreferences using skip-mention sequences.

A worked example

In this section we illustrate the detection of coreferences using our approach from the following document: "`John is married to Jena. He is a mechanic at OBI and she works there. It is a DIY market.`". Let $\overline{x} = [x_1, x_2, x_3, ..., x_n]$ denote a sequence of all mentions within the document. Mentions x_i are ordered by their occurrence in the document. For example, from the document we select all entity mentions into one training mention sequence x:

$$x = [\texttt{John}, \texttt{Jena}, \texttt{He}, \texttt{OBI}, \texttt{she}, \texttt{there}, \texttt{It}, \texttt{DIY market}] \tag{3}$$

As mentions mostly consist of noun phrases we could also identify `a mechanic` as a mention. Due to the simplification of the process the phrase was not identified as a mention during the mention detection. Our goal is now to detect the target clusters for each entity $x_{\texttt{John}}, x_{\texttt{Jena}}$ and $x_{\texttt{OBI}}$:

$$x_{\texttt{John}} = \{\texttt{John}, \texttt{He}\}, \tag{4}$$

$$x_{\texttt{Jena}} = \{\texttt{Jena}, \texttt{she}\}, \tag{5}$$

$$x_{\texttt{OBI}} = \{\texttt{OBI}, \texttt{there}, \texttt{It}, \texttt{DIY market}\} \tag{6}$$

In some cases, a mention could overlap with another mention. We treat such pairs as separate mentions and order them lexicographically by the index of the first word and mention length.

First, we decide to use zero, one and two for s skip-mention sequences and this is also a parameter to the system. In Figure 3, we show a training mention sequence x, which is applicable to first-order probabilistic models. We call it a 'zero skip-mention sequence' because it includes all mentions from a document and there are no (i.e., zero) other mentions between any two consecutive mentions in it. To identify coreferent mentions in the sequences, we need to label them using the labels $\{O, C\}$. The label C states that the current mention is coreferent with the previous one, whereas O states that the current mention is not coreferent with the previous one. Our linear-chain CRF models are learned over these labels and are therefore able to infer new labels for unseen mention sequences. Observe that for the toy

example above, first-order models detect just three coreferent mentions $\{\text{there}, \text{It}, \text{DIY Market}\}$ from a zero skip-mention sequence.

To solve the problem of identification of coreferent mentions at longer distances that contain other mention in between (e.g., OBI and there), we introduce further transformations. All additional skip-mention sequences are generated from the initial zero skip-mention sequence x and are labeled accordingly using $\{O, C\}$ labels. We also train a separate linear-chain CRF model for each additional skip-mention sequence type, which enables us to tag new unseen data for specific skip-mention distance.

Next, we then generate one skip-mention sequences (see Figure 4), which contain every second mention from the x above. The trained model for one skip-mention sequences can therefore extend our results by two new pairs $\{\texttt{John}, \texttt{He}\}$ and $\{\texttt{OBI}, \texttt{there}\}$. Analogously, for the two skip-mention sequences (see Figure 5) we could get our final missing pairs $\{\texttt{OBI}, \texttt{It}\}$ and $\{\texttt{Jena}, \texttt{she}\}$.

Lastly, we perform mention clustering from the previously extracted results from all the skip-mention sequences and return target entity clusters $x_{\texttt{John}}, x_{\texttt{Jena}}$ and $x_{\texttt{OBI}}$.

As shown in the example above, the transformation into higher skip-mention sequences returns more sequences per document. Intuitively, at distance zero, we get one training sequence per document (it contains all document mentions). At distance one, we get two sequences (each contains every second mention). At distance two, we get three sequences, etc. Therefore, the transformation into d skip-mention sequences returns $d + 1$ sequences of length $\lceil \frac{n}{d} \rceil$, where n is the number of all mentions in the document.

The SkipCor System

The SkipCor system takes a set of documents as input and returns a set of coreferent mention clusters, where each cluster represents an entity to which the mentions refer. The algorithm first reads mentions from the text and then transforms them into skip-mention sequences. Then, we load LCRF models specific to the generated skip-mention sequences and each of these independent models returns separately tagged skip-mention sequences, which are used at the clustering step. The final result is therefore a set of entities (represented as clusters of mentions) for each input document. We show a high level SkipCor data flow in Figure 6 and the detailed algorithms for training and inference are presented in Table 2 and Table 3, respectively.

The training phase is similar to the inference phase. The only difference is that the training must occur before any inference (the dashed rectangle in Figure 6). Each of the trained LCRF models is then able to infer the labels for a specific skip-mention distance.

During the training phase, Table 2, we build a skip-mention coreference resolution model. The algorithm takes as input the training documents, a list of feature functions, and a list of skip-mention distances. First, in the pre-processing step, we import the training data in the form of sentences and enrich them with additional tags (e.g., part-of-speech tags, lemmas, parse trees). Then we generate mention sequences (i.e., with zero skip-mentions) for each document. These sequences contain references to the original sentences, therefore the feature functions can use context data from the original input text and not only from the mention sequences. The main part of training the algorithm is the for loop, in which we transform the original mention sequences into the appropriate s_i skip-mention sequences, generate features, and train a specific model for every s_i using the `LCRFTrain` function. Each for loop execution is independent of the others, thus, the algorithm can be parallelized. Lastly, the final result of training is a SkipCor model, which is a tuple consisting of a list of trained skip-mention linear-chain CRF models, a list of the corresponding skip-mention distances, and a list of the feature functions.

To detect coreferences in unseen documents, we follow the algorithm shown in Table 3. As input, we take a raw text document and a SkipCor model that was trained using the algorithm in Table 2. During the execution, similarly to the training phase, we preprocess the input document and generate the initial mention sequence. If the mentions were not already detected in the input document, we perform a rule-based mention detection [7] to generate the initial mention sequence. Due to fact that we are processing

only one document, we get only one zero skip-mention sequence at this step (line 2). In the parallel for loop, we transform the initial mention sequence into s_i skip-mention sequences, generate the features, and execute the labeling of the specific s_i skip-mention LCRF model. All mention pairs that are identified as coreferring are stored in a set, which is the result of the parallel for loop. Lastly, during the clustering step we merge the coreferent mentions into mention clusters, where each cluster represents an underlying entity. These entity clusters are returned as the final result of the SkipCor coreference resolution.

The clustering step is performed using hierarchical agglomerative clustering. All the identified coreferent pairs that were extracted from the labeled zero-skip mention sequence are represented as initial mention clusters. If a mention is coreferent to no other mentions, it will form a singleton cluster. The initial clusters are then iteratively merged according to other labeled s_i skip-mention sequences. The final result of clustering is also the final result of the SkipCor labeling, and consists of a set of clusters that represent separate entities.

The time complexity of both proposed methods is mainly determined by the training and inference of the LCRF models (i.e., `LCRFTrain` and `LCRFLabel`), since other routines can be run in linear time. Still, some third-party methods used at pre-processing could consume more time. Due to the parallel execution of the for loop, we need to find the longest lasting execution. Let us say that the CRF training or inference has a time complexity of $O(EL^Q)$ [40], where E is the number of edges in the graph, L is the number of labels, and Q is the size of the maximal clique. In our type of CRF model, we use two possible labels: O, C, and the size of every clique is two. The number of edges E depends on the sequence input to the algorithm. Let us say that there are n mentions in a document, which results in a zero skip-mention sequence with $2n - 1 = O(n)$ edges. Moreover, every other generated d skip-mention sequence contains $d(\lceil \frac{2n}{d} \rceil - 1) = 2n - d = O(n)$ edges. Thus, we conclude that by employing parallelization, CRF models would use $O(2^2 n) = O(n)$ of time. Additionally, next to other linear time procedures, it is also important to include the time for feature function initialization, which takes on the order of $O(nm)$, where m is the number of input feature functions.

Results and Discussion

In this section, we first explain the coreference resolution evaluation metrics, the system settings that are used during the analysis, and give an overview of the SkipCor baseline systems. Then we introduce the evaluation datasets with some general statistics, labeling specifics, and additional attributes used for training. Next, we show the evaluation results on all the datasets, compare the SkipCor system to two baseline systems, and discuss the results. Lastly, we see how the system accuracy drops when training it on one dataset and testing it on another, to show the expected accuracy in real life scenarios.

Experimental Framework

There is no general agreement on which metric to use for the coreference resolution task. We here adopt the measures most commonly used in the literature, which will be described below. Prior to the measures we use in this paper, a graph based scoring algorithm had been used, that produced very unintuitive results [41, 42]. There have been a number of metrics proposed, so we evaluate the system using the following most commonly used measures:

MUC The key idea in developing the MUC measure [17] 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 in order to connect all the mentions referring to an entity. Recall, on the other hand, 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

having more mentions per entity, while it also ignores entities with only one mention (singleton entities).

BCubed The BCubed metric [43] 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 [44] is to achieve better interpretability. The result therefore 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 $\frac{\text{total similarity}}{|r|}$.

For the evaluation in this paper, only exact mention matches are considered as correct, see [45] with some modifications proposed by Cai and Strube [4].

The majority of the state-of-the-art systems were evaluated on specialized shared tasks at MUC (Message Understanding Conference) [46], ACE (Automatic Content Extraction) [47], SemEval2010 (Semantic Evaluation) [48], and, most recently, at CoNLL-2011 and CoNLL-2012 (Conference on Computational Language Learning) [45,49]. Some general information regarding the English datasets that we used in our evaluation is shown in Table 4. We focused primarily on newswire and broadcast news texts, which have been the most thoroughly studied in the past. To be more specific, we used the following datasets: (1) The ACE 2004 dataset, which in addition to broadcast news and newswire texts, also contains transcripts of conversations and various news reports transcribed and translated from Chinese and Arabic. It is the de facto standard dataset for all major information extraction tasks. (2) The SemEval 2010 dataset was designed specifically to evaluate coreference resolution systems in six languages. The English section of the dataset contains newswire and broadcast news from The Wall Street Journal and the TDT-4 collection. (3) The CoNLL 2012 corpus is one of the largest coreference resolution datasets. It tries to provide a much larger selection of coreferring entities, connecting together events and entities. The corpus consists of newswire texts, magazine articles, broadcast news, broadcast conversations, web data, conversational speech, and an English translation of the New Testament.

The proposed system is trained to detect coreferences over all tagged mention types: named, nominal, and pronominal. Due to differences in annotator agreements and rules for tagging the mentions, we cannot compare the results between the corpora. For example, the ACE and CoNLL datasets both include tags for all three mention types, but CoNLL includes more general entities. The CoNLL dataset also includes exact mention phrase boundaries, without considering parse tree constituents (a subtree that identifies an exact token sequence). Therefore it is expected for the results to be lower on CoNLL. Furthermore, SemEval includes only nominal mention types and heuristically identified singleton mentions. Nevertheless, we still conducted additional experiments involving training on one dataset type or domain and testing on another. We will present these results since the main motivation for the whole IE field is to develop techniques that work on an unpredictable user text input, where a user does not know what kind of data the algorithms were trained on.

To get additional annotations for the datasets, we used Apache OpenNLP toolkit [50] sentence splitter, POS tagger, and a dependency parser. For the LCRF training and inference, we used CRFSuite [51] with a cut-off threshold of three features and a default setting, which uses the L-BFGS optimization method. The whole implementation along with the evaluation of the proposed skip-mention coreference resolution is available in a public source code repository [33].

Empirical Comparisons

As already mentioned, the accuracy of the system depends on the skip-mention sequence types: the accuracy may not increase when using larger and larger skip-mention distances. In Figure 7, we show the results of training the models using different skip-mention sequence distances. From the results, we observe that when taking into account skip-mention distances larger than 40, the F1 scores do not increase or change significantly because although the recall scores increase, the precision scores decrease. Therefore, the final F1 score remains almost stalled due to a compensation of both scores, or even starts to slightly decrease. The scores that we further present in the evaluation were recorded using all skip-mention distances from zero to 25 (cut-off lines in Figure 7). We did not perform any mention detection, and therefore we always compare the results to the settings with already detected mentions.

We compared the proposed SkipCor system to the baseline systems SkipCorZero and SkipCorPair, both using the same feature functions and settings as SkipCor. The only difference between them is the use of different skip-mention sequence types. SkipCorZero detects coreferences only over zero skip-mention sequences, while SkipCorPair checks every mention pair within a document and predicts whether the two mentions are coreferent or not. Due to the large number of mention pairs considered by SkipCorPair, we limited the distance of the mention pairs to ten mentions. SkipCorPair therefore consists of ten LCRF models, each of which is trained to label coreferentiality on skip-mention sequences of length of two mentions.

In Table 5 we present the results for the ACE2004 dataset. When using the newswire and broadcast news portion, we split the data into training and testing sets in the ratio 70:30. For the whole ACE dataset, we used 336 documents for training and the others for testing [9]. SkipCorZero and SkipCorPair achieved relatively good or best precision values but very low recall. Generally, SkipCorPair outperformed SkipCorZero, while the proposed SkipCor system outperformed both of them. In comparison to other proposed systems, SkipCor achieved a slightly better BCubed score but a lower MUC score. As the results are so close, and opposite for the two measures, it is hard to decide which system is better. On broadcast news, we achieved better MUC and BCubed scores, which are similar to the ones from the newswire section. On the other hand, the precision values are lower, but we achieved a lower difference between the precision and recall compared to the competitive systems. Therefore, we uncovered a lot more mention clusters that have more errors, but the overall results are better. Lastly, we tested the system over the whole dataset (ACE2004-ALL), where we achieved results comparable to those of other systems.

The results for the CoNLL2012 dataset are shown in Table 6. The corpus is already separated into training, testing, and development datasets (we did not use the last when training). We used gold mention boundaries and additional manual tags, which are included in the data, therefore the results are comparable to the Gold Mention Boundaries setting. Fernandes et al. [36] proposed the shared task winning system and they are also the only ones who published their results on the broadcast news and newswire subdatasets (i.e., CONLL2012-BN, CONLL2012-NW). Similarly to the ACE2004 results, SkipCor performed better than SkipCorPair and SkipCorZero, except on the CoNLL2012-BN subdataset, where SkipCorPair outperformed SkipCor as it achieved the best precision and good recall. Otherwise, on most of the measures, SkipCor slightly outperformed the other systems and achieved better results with the MUC metric, generating cleaner mention clusters. For the full shared task, nine research teams submitted their results, but we show the results of only the top six. We significantly outperformed the others according to the MUC metric, where we increased the precision while having a comparable level of recall. According to the BCubed metric, the results are very similar, but in terms of CEAF we performed a little worse. The systems at the shared task were ranked using the CoNLL2012 measure, which is an average score of the MUC, the BCubed, and the CEAF F-scores. The winning Fernandes et al. [36] system achieved a CoNLL2012 score of 63.1 on English data, whereas our system achieved a score of 61.3, ranking as the second. The next then got the score of 60.7, with the others ranging down to the score of 43.0.

In Table 7 we show the results for the SemEval2010 dataset, which is already separated into training and testing portions. We compared the systems using the Gold-standard Closed setting, for which systems can use only the provided attributes with true mention boundaries. On this dataset, SkipCor outperformed SkipCorZero on all three measures and outperformed SkipCorPair in terms of the CEAF and BCubed metrics. Interestingly, SkipCorPair achieved a significantly higher MUC precision score, and it therefore outperformed SkipCor in this measure. Compared to other systems, SkipCor achieved better BCubed and CEAF scores, but a lower MUC score. Interestingly, in the selected setting, the RelaxCor system performed the best, but our system outperformed it on all three measures. Focusing only on the MUC measure, we got the second place, as the SUCRE system achieved a better recall score.

System UBIU [52], which entered the SemEval2010 shared task, also competed at the CoNLL2012 task, with a few modifications [53]. Our system significantly outperformed UBIU on both tasks and in terms of all three metrics. In contrast to our proposal, UBIU uses pairwise classification with a form of memory-based machine learning.

According to the results we showed, SkipCor outperformed both SkipCorZero and SkipCorPair. SkipCorZero mostly achieved good precision but very low recall. This is due to the identification of coreferences only between consecutive mentions within a document. SkipCor therefore uses skip-mention sequences to boost the recall values and consequently also the final result. SkipCorPair ranks somewhere between SkipCorZero and SkipCor. It checks for coreferences between mention pairs and is therefore very similar to other pairwise approaches. Due to a lot of pairwise comparisons, many mention sequences of length two must be generated, and therefore SkipCorPair executes more slowly than SkipCor.

Generally, SkipCor showed improvements on most of the datasets or achieved comparable results. We did not asses statistical significance of the differences in accuracy between the various systems because their implementations are not accessible and also referenced papers report single F score values only. Although some of the the existing rule-based systems are easy to implement and achieved good or best results, they may not be easily adapted to a different domain. This is also the reason why we proposed a simple machine-learned method for the task. SkipCor mostly obtained very good recall scores and a little bit lower precision. Other top performance systems use hybrid approaches, combining rule-based strategies with machine learning. All of them also employ feature engineering with a heavy use of lexicalized features. At the ACE2004 task, Haghighi et al. [13] used a completely deterministic approach, driven entirely by syntactic and semantic constraints. Bengston and Roth [14] focused especially on rich feature functions engineering with a simple pairwise classifier based on averaged perceptron. At the SemEval2010 shared task, the best two systems used a combination of manual rules and a set of machine learning classifiers (i.e., decision trees, naive Bayes, SVM, or maximum entropy models). Lastly, the CoNLL2012 task winner, Fernandes et al. [36], looked for the best mention clustering within a document using a specialized version of structure perceptron and represented mention clusters as coreference trees. The only system that used first-order probabilistic models was the one by Cullota et al. [9] on the ACE dataset. Their usage is completely different from that of our approach, because they still perform standard pairwise comparisons and then use first-order logic over mention clusters. Other CRF-based approaches, which were mentioned within the section on related work, were tested only against a limited version of a coreference resolution dataset or focused on an entity resolution task, which is a little similar to coreference resolution.

Performance in Real-world Scenarios

In addition to standard evaluation techniques, we trained SkipCor on one dataset and tested it on another (Table 8). Although the datasets do not have the same annotation guidelines or domain, this is interesting, as showing the results that can be expected by an end user on real data.

First, we notice only a minor performance drop when testing within the datasets from the same shared task. For example, the results between broadcast news and newswire data remained almost the same as for the CoNLL and ACE2004 data separately. Furthermore, CoNLL models performed only a little

worse on the ACE2004 dataset than originally. On the other hand, ACE2004 models performed less well on the CoNLL dataset, with a drop of roughly 20%. Both the CoNLL and ACE2004 models achieved low MUC scores on SemEval, but the best BCubed and CEAF scores. The difference is due to the fact that SemEval contains only nominal mentions and heuristically tagged singletons, which are more easily discovered, and they boost the scores. A model trained on SemEval performed the worst on both CoNLL and ACE2004. Interestingly, it achieved better MUC scores on CoNLL data than on the native SemEval testing dataset.

To conclude, the results typically show drops in accuracy on other domains or other datasets of the same or a different domain, from their performance on the same dataset. A similar analysis on different coreference datasets has also been conducted before [54], and their findings also show that evaluation on the same dataset the models were trained on gives the best results.

Conclusions

The present paper proposed 'SkipCor', a novel skip-mention coreference resolution system that is based solely on the linear-chain conditional random fields algorithm. To support the identification of all coreferent mentions in the text, the basic algorithm was extended with an adequate transformation of the data into different skip-mention sequences. In contrast to traditional approaches, the proposed system avoids checking all possible pairwise comparisons or using a single model. Thus, the system is completely parallelizable with a linear time complexity. Due to the amount of textual data available to date, the latter is of considerable importance in practical applications. We also stressed that the proposed skip-mention sequences could be adopted within other approaches in a straightforward fashion, which represents a prominent direction for future research.

The proposed system was evaluated on standard coreference resolution datasets that are the focus of evaluations for the majority of the techniques in the field. We compared the system to some baseline algorithms and also to the best performing coreference systems reported in the literature. The results obtained are comparable to the current state-of-the-art in coreference resolution, while we also more thoroughly analysed the contribution of the proposed skip-mention sequences. In addition, the analysis revealed that although accuracy in real-world scenarios can be even larger than expected, it decreases significantly when the system is trained on less reliable datasets.

Future work will focus on the development of more intelligent SkipCor mention clustering techniques (e.g., weighted scoring of coreference models) to minimize the number of merged conflicting mentions. Moreover, the system will be extended with a domain ontology that will provide an additional source of feature functions.

Acknowledgments

References

1. Andersen PM, Hayes PJ, Huettner AK, Schmandt LM, Nirenburg IB, et al. (1992) Automatic extraction of facts from press releases to generate news stories. In: Proceedings of the third conference on Applied natural language processing. Pennsylvania: Association for Computational Linguistics, pp. 170-177.

2. Sarawagi S (2008) Information extraction. Foundations and Trends in Databases 1: 261–377.

3. Wimalasuriya DC, Dou D (2010) Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science 36: 306–323.

4. Cai J, Strube M (2010) End-to-end coreference resolution via hypergraph partitioning. In: Proceedings of the 23rd International Conference on Computational Linguistics. Pennsylvania: Association for Computational Linguistics, pp. 143-151.

5. Ng V (2008) Unsupervised models for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Pennsylvania: Association for Computational Linguistics, pp. 640-649.

6. Luo X (2007) Coreference or not: A twin model for coreference resolution. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics. New York: Association for Computational Linguistics, pp. 73-80.

7. Lee H, Peirsman Y, Chang A, Chambers N, Surdeanu M, et al. (2011) Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. Pennsylvania: Association for Computational Linguistics, pp. 28-34.

8. McCarthy JF, Lehnert WG (1995) Using decision trees for coreference resolution. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence. Montreal, pp. 1050–1055.

9. Culotta A, Wick M, Hall R, McCallum A (2007) First-order probabilistic models for coreference resolution. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 81-88.

10. Ng V, Cardie C (2002) Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Philadelphia: Association for Computational Linguistics, pp. 104-111.

11. Cardie C, Wagstaff K (1999) Noun phrase coreference as clustering. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Pennsylvania: Association for Computational Linguistics, pp. 82-89.

12. Raghunathan K, Lee H, Rangarajan S, Chambers N, Surdeanu M, et al. (2010) A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Pennsylvania: Association for Computational Linguistics, pp. 492-501.

13. Haghighi A, Klein D (2009) Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Pennsylvania: Association for Computational Linguistics, volume 3, pp. 1152-1161.

14. Bengtson E, Roth D (2008) Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Pennsylvania: Association for Computational Linguistics, pp. 294-303.

15. Clark JH, González-Brenes JP (2008) Coreference resolution: Current trends and future directions. Language and Statistics II Literature Review : 1–14.

16. Bernardi R, Kirschner M, Ratkovic Z (2010) Context fusion: The role of discourse structure and centering theory. In: Proceedings of The International Conference on Language Resources and Evaluation. Valletta: European Language Resources Evaluation, pp. 1–8.

17. Vilain M, Burger J, Aberdeen J, Connolly D, Hirschman L (1995) A model-theoretic coreference scoring scheme. In: Proceedings of the 6th conference on Message understanding. Pennsylvania: Association for Computational Linguistics, pp. 45-52.

18. Nguyen N, Kim JD, Miwa M, Matsuzaki T, Tsujii J (2012) Improving protein coreference resolution by simple semantic classification. BMC Bioinformatics 13: 1–21.

19. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, pp. 282-289.

20. Soon WM, Ng HT, Lim DCY (2001) A machine learning approach to coreference resolution of noun phrases. Computational linguistics 27: 521-544.

21. Luo X, Ittycheriah A, Jing H, Kambhatla N, Roukos S (2004) A mention-synchronous coreference resolution algorithm based on the bell tree. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Pennsylvania: Association for Computational Linguistics, pp. 136–143.

22. Rahman A, Ng V (2009) Supervised models for coreference resolution. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. volume 2, pp. 968-977.

23. Huang S, Zhang Y, Zhou J, Chen J (2009) Coreference resolution using markov logic networks. Advances in Computational Linguistics 41: 157–168.

24. Stoyanov V, Cardie C, Gilbert N, Riloff E, Buttler D, et al. (2010) Coreference resolution with reconcile. In: Proceedings of the Association for Computational Linguistics 2010 Conference - Short Papers. Pennsylvania: Association for Computational Linguistics, pp. 156-161.

25. Bejan C, Titsworth M, Hickl A, Harabagiu S (2009) Nonparametric bayesian models for unsupervised event coreference resolution. In: Advances in Neural Information Processing Systems. pp. 73-81.

26. Bejan CA, Harabagiu S (2010) Unsupervised event coreference resolution with rich linguistic features. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, pp. 1412-1422.

27. Charniak E (2001) Unsupervised learning of name structure from coreference data. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. Stroudsburg: Association for Computational Linguistics, pp. 1–7.

28. McCallum A, Wellner B (2004) Conditional models of identity uncertainty with application to noun coreference. In: Neural Information Processing Systems. pp. 1–8.

29. Fosler-Lussier E, He Y, Jyothi P, Prabhavalkar R (2013) Conditional random fields in speech, audio, and language processing. Proceedings of the IEEE 101: 1054–1075.

30. Wellner B, McCallum A, Peng F, Hay M (2004) An integrated, conditional model of information extraction and coreference with application to citation matching. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. Arlington: AUAI Press, pp. 593-601.

31. Finkel JR, Grenager T, Manning C (2005) Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Pennsylvania: Association for Computational Linguistics, pp. 363-370.

32. Ram RVS, Devi SL (2012) Coreference resolution using tree CRFs. In: Proceedings of Conference on Intelligent Text Processing and Computational Linguistics. Heidelberg: Springer, pp. 285-296.

33. Žitnik S (2013). Intelligent Ontology-based Information Extraction - IOBIE, source code repository. Available from `https://bitbucket.org/szitnik/iobie` (last accessed 15th May 2014).

34. Broscheit S, Poesio M, Ponzetto SP, Rodriguez KJ, Romano L, et al. (2010) BART: a multilingual anaphora resolution system. In: Proceedings of the 5th International Workshop on Semantic Evaluation. Pennsylvania: Association for Computational Linguistics, pp. 104-107.

35. Attardi G, Rossi SD, Simi M (2010) TANL-1: coreference resolution by parse analysis and similarity clustering. In: Proceedings of the 5th International Workshop on Semantic Evaluation. Pennsylvania: Association for Computational Linguistics, pp. 108-111.

36. Fernandes ER, dos Santos CN, Milidiú RL (2012) Latent structure perceptron with feature induction for unrestricted coreference resolution. In: Proceedings of CoNLL 2012 Joint Conference on EMNLP and CoNLL. Pennsylvania: Association for Computational Linguistics, pp. 41–48.

37. Miller GA (1995) WordNet: a lexical database for english. Communications of the ACM 38: 39-41.

38. Bergsma S, Lin D (2006) Bootstrapping path-based pronoun resolution. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Pennsylvania: Association for Computational Linguistics, pp. 33-40.

39. Orasan C, Evans R (2007) NP animacy identification for anaphora resolution. Journal of Artificial Intelligence Research 29: 79-103.

40. Cohn T (2006) Efficient inference in large conditional random fields. In: Proceedings of the 17th European conference on Machine Learning. Heidelberg: Springer, pp. 606-613.

41. Chincor N (1991) MUC-3 evaluation metrics. In: Proceedings of the 3rd conference on Message understanding. Pennsylvania: Association for Computational Linguistics, pp. 17–24.

42. Chinchor N, Sundheim B (1993) MUC-5 evaluation metrics. In: Proceedings of the 5th conference on Message understanding. Pennsylvania: Association for Computational Linguistics, p. 6978.

43. Bagga A, Baldwin B (1998) Algorithms for scoring coreference chains. In: The first international conference on language resources and evaluation workshop on linguistics coreference. volume 1, pp. 1–7.

44. Luo X (2005) On coreference resolution performance metrics. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Pennsylvania: Association for Computational Linguistics, pp. 25-32.

45. Pradhan S, Moschitti A, Xue N, Uryupina O, Zhang Y (2012) CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In: Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task. Association for Computational Linguistics, pp. 1–40.

46. Hirschman L, Chincor NA (1997) MUC-7 coreference task definition. In: Proceedings of the Seventh Message Understanding Conference. San Francisco: Morgan Kaufmann, pp. 1–17.

47. Doddington G, Mitchell A, Przybocki M, Ramshaw L, Strassel S, et al. (2004) The automatic content extraction (ACE) programtasks, data, and evaluation. In: Proceedings of LREC. volume 4, pp. 837-840.

48. Recasens M, Màrquez L, Sapena E, Mart AM, Taulé M, et al. (2010) SemEval-2010 task 1: Coreference resolution in multiple languages. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 1–8.

49. Pradhan S, Ramshaw L, Marcus M, Palmer M, Weischedel R, et al. (2011) CoNLL-2011 shared task: Modeling unrestricted coreference in ontonotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. pp. 1-27.

50. Apache OpenNLP: a machine learning based toolkit for the processing of natural language text. Available from `http://opennlp.apache.org/` (last accessed 15th May 2014).

51. Okazaki N (2007). CRFsuite: a fast implementation of conditional random fields (CRFs). Available from `http://www.chokkan.org/software/crfsuite` (last accessed 15th May 2014).

52. Zhekova D, Kübler S (2010) UBIU: a language-independent system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 96-99.

53. Zhekova D, Kübler S, Bonner J, Ragheb M, Hsu YY (2012) UBIU for multilingual coreference resolution in OntoNotes. In: Joint Conference on EMNLP and CoNLL-Shared Task. pp. 88-94.

54. Gilbert N, Riloff E (2013) Domain-specific coreference resolution with lexicalized features. Pennsylvania: Association for Computational Linguistics, pp. 1–6.

55. Finkel JR, Manning CD (2008) Enforcing transitivity in coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. pp. 45-48.

56. Björkelund A, Farkas R (2012) Data-driven multilingual coreference resolution using resolver stacking. In: Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task. pp. 49–55.

57. Chen C, Ng V (2012) Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In: Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task. pp. 56–63.

58. Stamborg M, Medved D, Exner P, Nugues P (2012) Using syntactic dependencies to solve coreferences. In: Joint Conference on EMNLP and CoNLL-Shared Task. pp. 64-70.

59. Li X, Wang X, Liao X (2012) Simple maximum entropy models for multilingual coreference resolution. In: Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task. pp. 83–88.

60. Sapena E, Padró L, Turmo J (2010) RelaxCor: a global relaxation labeling approach to coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 88-91.

61. Kobdani H, Schtze H (2010) SUCRE: a modular system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 92-95.

62. Winkler WE (1990) String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proceedings of the Section on Survey Research Methods. American Statistical Association, pp. 354–359.

63. Bansal M, Klein D (2012) Coreference semantics from web features. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. p. 389398.

Figure Legends

Tables

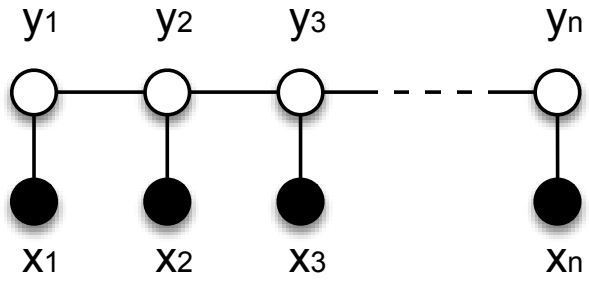

Figure 1. Linear-chain conditional random fields model. Black nodes represent observable values, which are in our case entity mentions. White nodes represent hidden labels that we need to predict and define whether the current observable value is coreferent with the previous one.

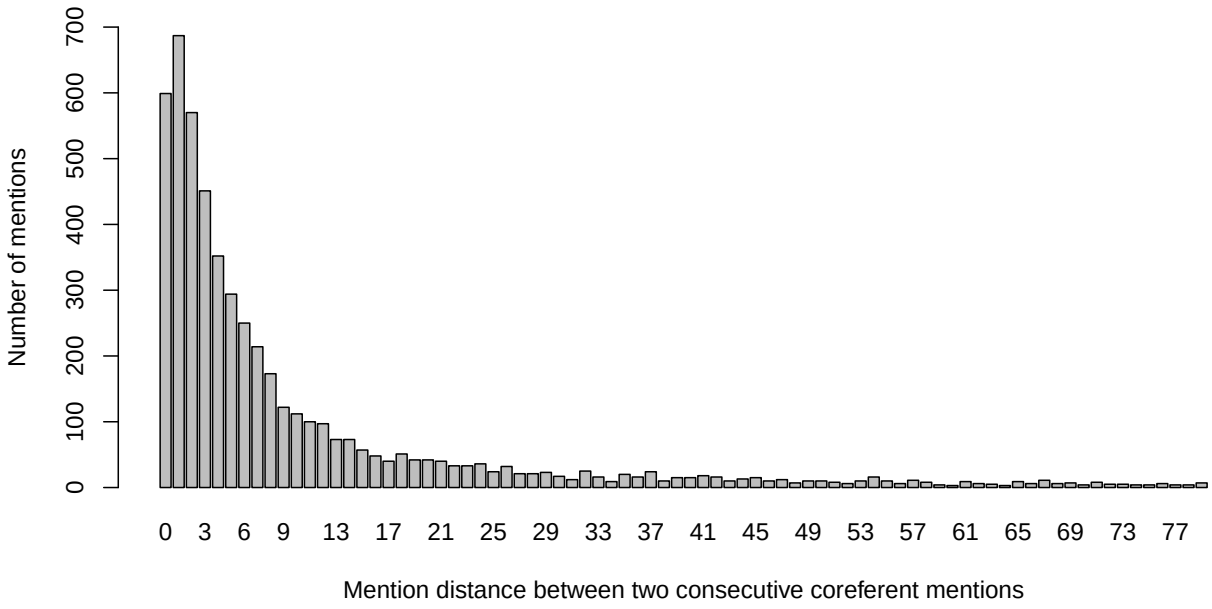

Figure 2. Distribution of distances between two consecutive coreferent mentions. The data was taken from the SemEval2010 [48] coreference dataset. Distance x between two consecutive mentions means that there exist x other mentions between them.

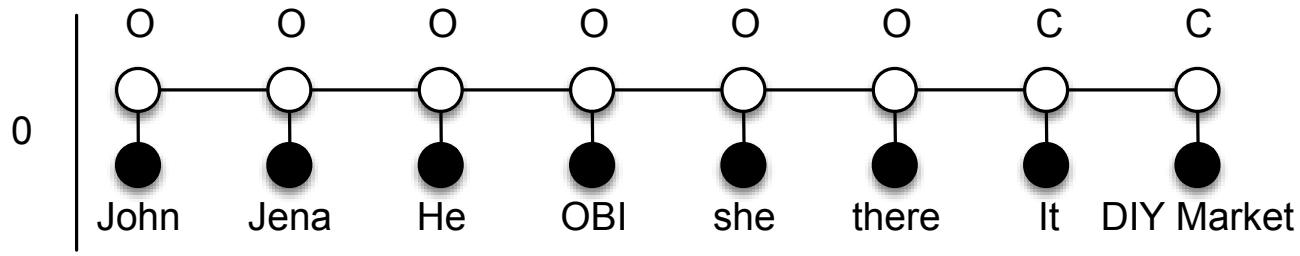

Figure 3. Zero skip-mention training sequence. Initial mention sequence that contains all mentions from the input text "*John* is married to *Jena*. *He* is a mechanic at *OBI* and *she* works *there*. *It* is a *DIY market*." If the current mention is coreferent with the previous one, it is labeled with *C*, otherwise with *O*.

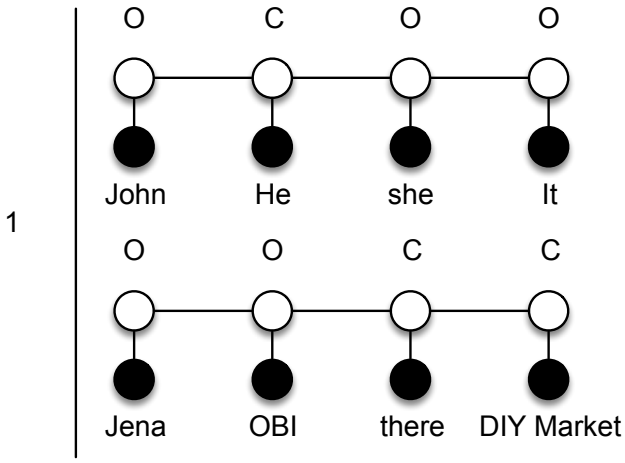

Figure 4. One skip-mention training sequences. Mention sequences that include every second mention (i.e., one skip-mention) from the input text "*John* is married to *Jena*. *He* is a mechanic at *OBI* and *she* works *there*. *It* is a *DIY market*." If the current mention is coreferent with the previous one, it is labeled with *C*, otherwise with *O*.

Table 1. Classification of coreference resolution approaches.

	UNSUPERVISED	**SUPERVISED**
SEQUENCE-BASED	[25–27]	**SkipCor**
PAIRWISE	[7, 12, 13], etc.	[9, 10, 14, 20–23, 28, 32]

According to the two-dimensional classification of coreference resolution systems, the proposed SkipCor system solves the problem in a novel fashion.

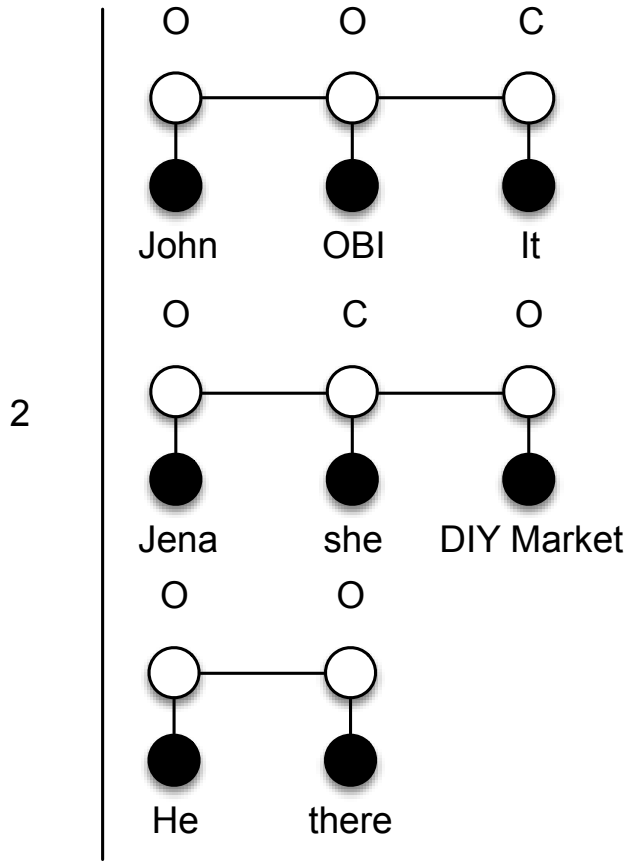

Figure 5. Two skip-mention training sequences. Mention sequences that include every third mention (i.e., two skip-mention) from the input text "*John* is married to *Jena*. *He* is a mechanic at *OBI* and *she* works *there*. *It* is a *DIY market*." If the current mention is coreferent with the previous one, it is labeled with *C*, otherwise with *O*.

Table 2. Algorithm 1.

Algorithm 1: **Skip-mention classifier training**
Input: training documents D, feature functions $f_l \in F$ and skip-mention distances $s_i \in S$
Output: skip-mention model $(skipMentionCRF, S, F)$
1: $sentences \leftarrow$ importTrainingData(D)
2: $sentences \leftarrow$ preprocessInputText$(sentences)$
3: $mentionSequences \leftarrow$ readMentions$(sentences)$
4: $skipMentionCRF \leftarrow$ [] //empty list
5: **parallel for each** $s_i \in S$:
6: $\quad skipMentionSequences \leftarrow$ transform$(mentionSequences, s_i)$
7: $\quad$ initializeFeatureFunctions$(skipMentionSequences, F)$
8: $\quad skipMentionCRF_i \leftarrow$ LCRFTrain$(skipMentionSequences)$
9: **return** $(skipMentionCRF, S, F)$

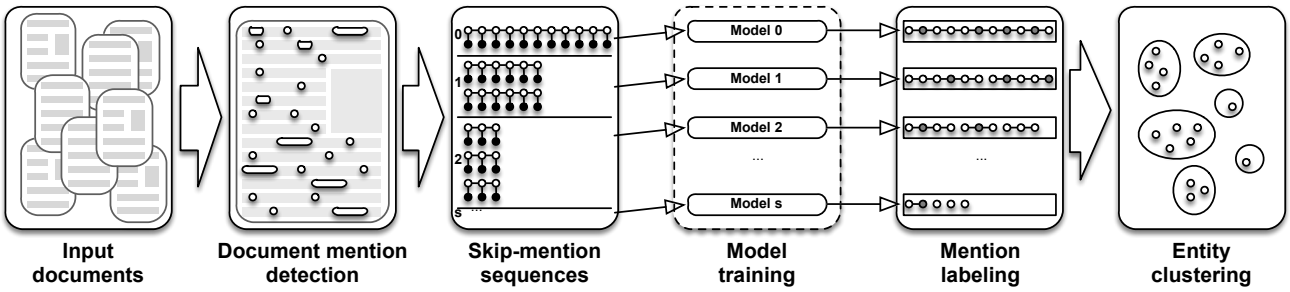

Figure 6. High level skip-mention coreference resolution data flow. The input to the system is given as a set of documents. For each document we select mentions and transform them into mention sequences. According to the system parameters, sequences contain every *s+1*th mention (i.e., *s* skip-mention). A model is trained for each sequence type and then used for labeling. After sequences are labeled, the mentions are then clustered. Each cluster of mentions represents a specific entity, which is also the final result of the system.

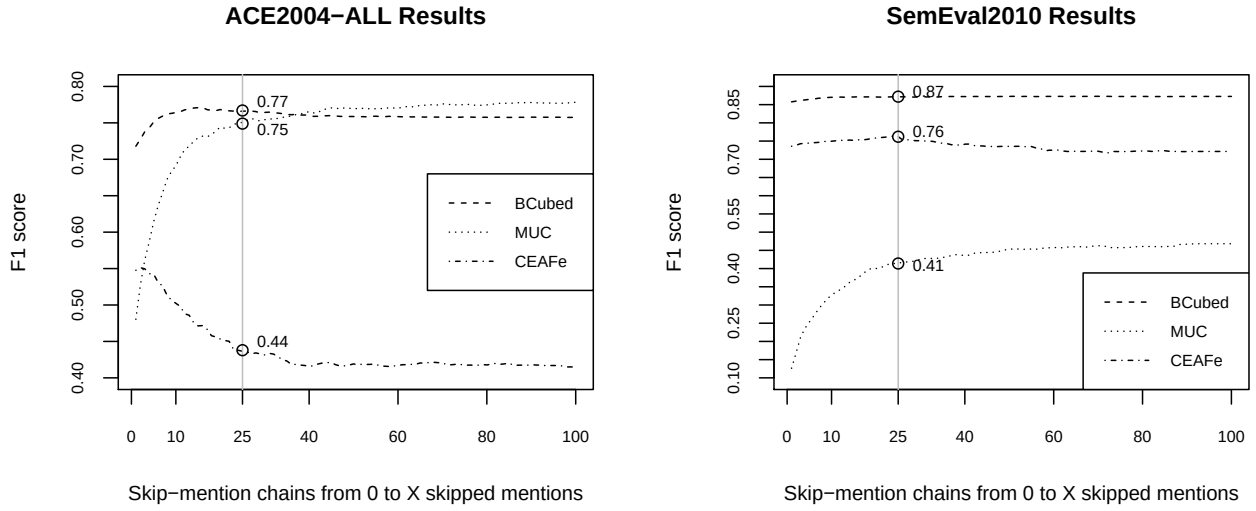

Figure 7. Coreference resolution results using different skip-mention sequences. Evaluation of the proposed system on the whole ACE2004 [47] and SemEval2010 [48] datasets using the metrics BCubed [43], MUC [17] and CEAFe [44].

Table 3. Algorithm 2.

Algorithm 2: **Skip-mention classifier labeling**

Input: document D and a skip-mention model $(skipMentionCRF, S, F)$
Output: coreferent mention clusters
1: $sentences \leftarrow$ preprocessInputText(D)
2: $mentionSequence \leftarrow$ detectMentions($sentences$)
3: $coreferentMentions \leftarrow \emptyset$
4: **parallel for each** $s_i \in S$:
5: $skipMentionSequences \leftarrow$ transform($mentionSequence, s_i$)
6: initializeFeatureFunctions($skipMentionSequences, F$)
7: $coreferentMentions$ add LCRFLabel($skipMentionCRF_i, skipMentionSequences$)
8: $mentionClusters \leftarrow$ cluster($coreferentMentions$)
9: **return** $mentionClusters$

Table 4. Dataset descriptions.

Dataset	# documents	# sentences	# tokens	# mentions	# entities
ACE2004-ALL	450	7,518	191,387	29,724	12,439
ACE2004-NW	127	2,865	74,987	11,188	4,701
ACE2004-BN	220	3,782	71,602	11,323	4,918
SemEval2010-Train	229	3,648	78,831	21,550	16,082
SemEval2010-Test	85	1,141	24,121	6,692	4,839
CoNLL2012-ALL-Train	1,914	75,185	1,299,310	154,760	33,113
CoNLL2012-ALL-Test	221	9,479	169,579	19,677	4,217
CoNLL2012-NW-Train	734	15,288	387,082	34,470	9,404
CoNLL2012-NW-Test	88	1,898	49,235	4,361	1,168
CoNLL2012-BN-Train	748	9,723	180,300	22,262	6,433
CoNLL2012-BN-Test	93	1,252	23,209	2,936	790

The acronyms ALL (i.e., whole), NW (i.e., newswire), BN (i.e., broadcast news) stand for different subdatasets of the whole dataset, which is further divided into training and test portions.

Table 5. Results of the proposed SkipCor system, baseline systems, and other approaches on the ACE2004 datasets.

	MUC			BCubed		
	P	**R**	**F**	**P**	**R**	**F**
System	**ACE2004-NW**					
SkipCor	78.6	68.8	73.4	75.7	**78.6**	**77.1**
SkipCorZero	78.5	22.6	35.1	**96.3**	51.9	67.4
SkipCorPair	78.2	49.0	60.3	85.3	61.7	71.6
Finkel et al. [55]	78.7	58.5	67.1	86.8	65.2	74.5
Soon et al. [20][1]	**85.3**	37.8	52.4	94.1	56.9	70.9
Haghighi et al. [13]	77.0	**75.9**	**76.5**	79.4	74.5	76.9
Stoyanov et al. [24]	-	-	62.1	-	-	75.5
	ACE2004-BN					
SkipCor	76.3	**71.3**	**73.7**	76.2	**81.5**	**78.8**
SkipCorZero	79.3	28.3	41.7	**95.9**	57.3	71.8
SkipCorPair	80.9	59.4	68.5	86.3	70.7	77.7
Finkel et al. [55]	87.8	46.8	61.1	93.5	59.9	73.1
Soon et al. [20][1]	**90.0**	43.2	58.3	95.6	58.4	72.5
	ACE2004-ALL					
SkipCor	79.5	70.9	75.0	76.3	**81.1**	78.6
SkipCorZero	**81.3**	28.9	42.6	**95.6**	55.4	70.2
SkipCorPair	80.5	57.1	66.8	84.8	68.9	76.0
Cullota et al. [9]	-	-	-	86.7	73.2	79.3
Bengston et al. [14]	-	-	-	88.3	74.5	**80.8**
Haghighi et al. [13][2]	74.8	**77.7**	**76.2**	79.6	78.5	79.0

Coreference resolution systems evaluated on the ACE2004 dataset (i.e., ALL) [47] and its newswire (i.e., NW) and broadcast news (i.e., BN) subdatasets using the metrics MUC [17] and BCubed [43].
[1] Results were reported by Finkel and Manning [44].
[2] The MUC F1-score value does not agree with reported precision and recall and has been recalculated.

Table 6. Results of the proposed SkipCor system, baseline systems, and other approaches on the CoNLL2012 datasets.

	MUC			BCubed			CEAF		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
System	**CoNLL2012-NW**								
SkipCor	76.1	**61.5**	**68.0**	69.6	**70.9**	**70.2**	39.9	**59.8**	**47.8**
SkipCorZero	76.7	16.8	27.5	**95.5**	36.7	53.0	35.1	35.3	35.2
SkipCorPair	**81.3**	45.9	58.7	86.4	56.7	68.5	**42.0**	54.7	47.5
Fernandes et al. [36]	-	-	61.6	-	-	70.0	-	-	46.6
	CoNLL2012-BN								
SkipCor	76.2	**62.0**	**68.4**	69.8	**71.3**	**70.5**	34.2	**59.2**	43.3
SkipCorZero	76.8	19.4	31.0	**95.9**	45.8	62.0	37.4	37.6	37.5
SkipCorPair	**81.9**	44.7	57.8	88.6	54.6	67.6	**39.1**	56.3	46.1
Fernandes et al. [36]	-	-	65.6	-	-	70.0	-	-	**49.1**
	CoNLL2012-ALL								
SkipCor	84.9	63.6	**72.7**	74.6	65.6	69.8	32.9	**57.2**	41.7
SkipCorZero	81.8	21.7	34.3	**95.7**	39.7	56.1	32.1	32.7	32.4
SkipCorPair	**85.9**	50.7	63.8	86.5	53.4	66.0	30.8	53.8	39.2
Fernandes et al. [36]	77.5	64.9	70.7	79.0	64.3	**70.9**	**41.7**	56.5	**48.0**
Björkelund et al. [56]	71.6	63.4	67.3	76.6	64.0	69.7	41.4	50.0	45.3
Chen et al. [57]	66.8	63.3	65.0	73.6	65.4	69.2	44.9	48.8	46.8
Stamborg et al. [58]	58.8	**66.2**	62.3	65.0	**71.2**	68.0	45.8	38.5	41.8
Zhekova et al. [53]	54.7	55.0	54.8	55.6	61.9	58.6	34.7	34.4	34.5
Li et al. [59]	33.7	44.2	38.2	53.9	66.4	59.5	36.5	27.5	31.4

Coreference resolution systems evaluated on the CoNLL2012 dataset (i.e., ALL) [45], and its newswire (i.e., NW) and broadcast news (i.e., BN) subdatasets using the metrics MUC [17], BCubed [43] and CEAF [44].

Table 7. Results of the proposed SkipCor system, baseline systems, and other approaches on the SemEval2010 dataset.

	MUC			BCubed			CEAF		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
System	**SemEval2010**								
SkipCor	68.8	30.1	41.8	94.8	**80.8**	**87.3**	74.0	78.5	**76.2**
SkipCorZero	67.0	3.6	6.8	**99.6**	75.1	85.7	73.0	73.1	73.1
SkipCorPair	**76.7**	35.6	48.7	97.1	79.0	87.1	72.7	**79.4**	75.9
RelaxCor [60]	72.4	21.9	33.7	97.0	74.8	84.5	**75.6**	75.6	75.6
SUCRE [61]	54.9	**68.1**	**60.8**	78.5	86.7	82.4	74.3	74.3	74.3
TANL-1 [35]	24.4	23.7	24.0	72.1	74.6	73.4	61.4	75.0	67.6
UBIU [52]	25.5	17.2	20.5	83.5	67.8	74.8	68.2	63.4	65.7

Coreference resolution systems evaluated on the SemEval2010 dataset [48] using the metrics MUC [17], BCubed [43] and CEAF [44].

Table 8. Comparison of the results when training on one type of dataset or domain and testing on another.

	Model				
Dataset	**A-BN**	**A-NW**	**C-BN**	**C-NW**	**SemEval2010**
A-BN	*74, 78, 54*	**72**, 77, 39	65, 70, 28	64, 69, 29	42, **71**, 49
A-NW	**72**, 73, 42	*73, 75, 58*	60, 64, 27	59, 69, 29	42, 67, **50**
C-BN	33, 56, 37	40, 58, 39	*68, 70, 43*	**65**, 70, 27	**57**, 64, 31
C-NW	39, 57, 39	41, 59, 41	**67**, 66, 28	*68, 70, 48*	56, 64, 32
SemEval2010	19, **82**, **70**	23, **85**, **74**	39, 76, 40	39, **77**, **33**	*42, 87, 76*

Coreference resolution results comparison on ACE2004 (i.e., A), CoNLL2012 (i.e., C) and SemEval2010 newswire (i.e., NW) and broadcast news (i.e., BN) datasets. Each column represents a model trained on a specific dataset, while each row represents a dataset. Values represent F-scores of MUC [17], BCubed [43] and CEAF [44], respectively.

Table 9. Feature functions description.

Name	Description	Model
Target label distribution	Distribution of target labels.	A, S, C
Starts upper	Does the mention start with an upper case letter.	A, S, C
Starts upper twice	Do two consequent mentions start with an upper case letter.	A, S, C
Prefix value	Value of the prefix (length of 2 and 3) for the mention on offset distance (distances from -5 to 5) from the current mention.	A, S, C
Suffix value	Value of the suffix (length of 2 and 3) for the mention on offset distance (distances from -5 to 5) from the current mention.	A, S, C
Consequent value	A combination of values of the consequent mentions on offset distance (distances from -4 to 4) from the current mention.	A, S, C
String match	Do consequent mention values match.	A, S, C
Gender match	Does the gender of two consequent mentions match.	A, S, C
Gender value	The gender value of the mention.	A, S, C
Is appositive	Is the mention appositive of the another.	A, S, C
Alias	Is the mention alias or abbreviation of the another.	A, S, C
Is prefix	Is the mention prefix of the another.	A, S, C
Is suffix	Is the mention suffix of the another.	A, S, C
Similarity value	How similar are the two mention values according to the Jaro Winkler [62] metric.	A, S, C
Is pronoun	Is the mention a pronoun.	A, S, C
Same sentence	Are consequent mentions in the same sentence.	A, S, C
Hearst co-occurence [63]	Does the text between the two mentions follow some predefined rules, e.g. m_i such as m_j.	A, S, C
Sentence distance	What is the distance between the sentences of the two mentions.	A, S, C
Is quoted	Is the mention within the parentheses.	A, S, C
Substring match	Is the mention a substring of the another.	A
Starts with	Does the mention starts with the another.	A, S, C
Ends with	Does the mention ends with the another.	A, S, C
Number match	Do the mentions match in number (i.e., singular, plural).	A, S, C
Mention type	Type of mention (i.e., pronoun, name, nominal).	A
Relative pronoun	Heuristic decision if the mention is a relative pronoun of the another.	A
WordNet [37]	How is the mention semantically connected to the another (e.g., is a hypernym, synonym).	A
WordNet synset	Are the two consequent mentions in the same synset.	S, C
Entity type	What is the named entity type or subtype of the mention.	A
Length difference	What is the difference in length of the two consecutive mentions.	A, S, C
Is demonstrative	Is the mention a demonstrative noun phrase.	A, S, C
Offset match	Do consecutive POS values on distances from -2 to 2 match.	A
Parse tree path	Path values between the two mentions in a parse tree.	A, S, C
Parse tree mention depth	Depth of the mention within the parse tree.	A, S, C
Parse tree parent value	Parse tree value of the mention on lengths of one, two or three.	A, S, C
Relation	Does a relationship exist between the two consecutive mentions.	S
Speaker	Who is the current speaker in a transcript text.	C

The feature functions are used by all skip-mention CRF models and are modeled as unigram or bigram features. The exact details (e.g., which mention values are used by a specific feature functions) and implementations can be retrieved from our public source repository [33] (within the class `FeatureFunctionPackages`). The abbreviations A, S and C define which feature functions were used when training the models for the ACE2004, SemEval2010 and CoNLL2012 datasets, respectively.