Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 11–20
Marseille, 11–16 May 2020

A Cluster Ranking Model for Full Anaphora Resolution

Juntao Yu, Alexandra Uma, Massimo Poesio
Queen Mary University of London
{juntao.yu, a.n.uma, m.poesio}@qmul.ac.uk

Abstract

Anaphora resolution (coreference) systems designed for the CONLL 2012 dataset typically cannot handle key aspects of the full anaphora resolution task such as the identification of singletons and of certain types of non-referring expressions (e.g., expletives), as these aspects are not annotated in that corpus. However, the recently released CRAC 2018 Shared Task and Phrase Detectives (PD) datasets can now be used for that purpose. In this paper, we introduce an architecture to simultaneously identify non-referring expressions (including expletives, predicative NPs, and other types) and build coreference chains, including singletons. Our cluster-ranking system uses an attention mechanism to determine the relative importance of the mentions in the same cluster. Additional classifiers are used to identify singletons and non-referring markables. Our contributions are as follows. First of all, we report the first result on the CRAC data using system mentions; our result is 5.8% better than the shared task baseline system, which used gold mentions. Our system also outperforms the best-reported system on PD by up to 5.3%. Second, we demonstrate that the availability of singleton clusters and non-referring expressions can lead to substantially improved performance on non-singleton clusters as well. Third, we show that despite our model not being designed specifically for the CONLL data, it achieves a very competitive result.

Keywords: Anaphora Resolution, Coreference, Cluster ranking model, Non-referring detection, Deep Neural Networks

1. Introduction

Anaphora resolution is the task of identifying and resolving nominal anaphoric reference to discourse entities (Poesio et al., 2016b).[1] It is an important aspect of natural language processing and has a substantial impact on downstream applications such as summarization (Steinberger et al., 2007; Steinberger et al., 2016). Since the CONLL 2012 shared task (Pradhan et al., 2012), the ONTONOTES corpus has been the dominant resource in research on identity anaphora resolution (coreference) (Fernandes et al., 2014; Björkelund and Kuhn, 2014; Martschat and Strube, 2015; Clark and Manning, 2015; Clark and Manning, 2016a; Clark and Manning, 2016b; Lee et al., 2017; Lee et al., 2018; Kantor and Globerson, 2019; Joshi et al., 2019b; Joshi et al., 2019a). But ONTONOTES has a number of limitations. An often mentioned limitation is that singletons are not annotated (De Marneffe et al., 2015; Chen et al., 2018). A less discussed, but still crucial, limitation is that although some types of non-referring expressions are marked in ONTONOTES, in particular predicative ones (*a policeman* in *John is a policeman*), other types are not, such as expletives, meaning that in *It rained*, *It* is not considered a markable. As a consequence, systems optimized for ONTONOTES are only evaluated on non-singleton coreference chains; their performance at identifying singletons, and distinguishing them from expletives, is not evaluated.

But the decision to interpret *it* as referring or non-referring (Uryupina et al., 2016; Versley et al., 2008; Bergsma et al., 2008; Bergsma and Yarowsky, 2011; Hardmeier et al., 2015) is a key aspect of pronoun interpretation–for instance, for the purposes of machine translation (Guillou and Hardmeier, 2016)–so systems trained on ONTONOTES have had to adopt a variety of workarounds. These limitation of ONTONOTES have however been corrected in a number of corpora, including ANCORA for Spanish (Taulé et al., 2008), TUBA-D/Z for German (Telljohann et al.,), and, for English, ARRAU (Uryupina et al., 2019), which was used as dataset for the CRAC 2018 shared task (Poesio et al., 2018), and Phrase Detectives (PD) (Poesio et al., 2019).

The first contribution of this paper is the development of a system able to perform both coreference resolution and identification of non-referring markables and singletons, using the CRAC 2018 shared task and PD datasets. On CRAC, our model achieves a CONLL score of 77.9% on coreference chains, and an F1 score of 76.3% on non-referring expressions identification. This is, to the best of our knowledge, the first modern result on the CRAC data using system mentions. Our CONLL score is even 5.8% higher than the baseline result on this dataset, 72.1% obtained by (Poesio et al., 2018) using gold mentions. On PD, our model outperforms the best-performing system by up to 5.3%.

Our second contribution is a novel and competitive cluster ranking architecture for anaphora resolution[2]. Current coreference models can be classified either as mention pair models (Soon et al., 2001), in which connections are established between mentions, or entity mention models, in which mentions are directly linked to entities / coreference chains (Luo et al., 2004; Rahman and Ng, 2011). The mention pair models are simpler in concept and easier to implement, so many SoTA systems are exclusively based on mention ranking (Wiseman et al., 2015; Clark and Manning, 2016a; Lee et al., 2017). But it has long been known that entity-level information is important for coreference (Luo et al., 2004; Poesio et al., 2016b) so many systems attempted to explore features beyond those of mention pairs (Björkelund and Kuhn, 2014; Clark and Manning, 2015; Clark and Manning, 2016b; Lee et al., 2018; Kantor and Globerson, 2019; Joshi et al., 2019b; Joshi et al., 2019a). However, those systems are usually much more complex

[1] Some NLP researchers use the term anaphora resolution to refer to pronominal anaphoric reference only, but we use the term in the traditional linguistic and psycholinguistic sense (see (Poesio et al., 2016b) for full discussion).

[2] The code is available at https://github.com/juntaoy/dali-full-anaphora

than their mention ranking counterpart, since entity features are introduced in addition to their mention ranking part. Consider the Lee et al. (2018) system, for instance: the full system has 9.6 million trainable parameters in total, which is double the number of the mention ranking part of the system (4.8M parameters). In this work, we demonstrate that it is possible to achieve SoTA results by cluster ranking alone, i.e. by linking mentions directly to the entities. As a result, our model is less complex than the existing entity-level models (Lee et al., 2018; Kantor and Globerson, 2019) using similar mention representations. Our model uses only 4.8M trainable parameters without increasing the complexity of a mention ranking model. Furthermore, our model is fast to train; we show that a cluster ranking model can be significantly sped up by training on oracle clusters[3].

The key intuitions behind the proposed approach are (i) that cluster representations are crucial to the success of a cluster ranking system, and (ii) that a key property of these representations is that they should capture the fact that mentions in a cluster are not equally important. In particular, it is well-known that the mentions introducing an entity are generally more informative (e.g., *the president of ACME, John Smith*) whereas subsequent mentions tend to employ reduced forms (e.g., *Mr. Smith, he*) (Ariel, 1990). This motivates the use of cluster representations capable of preserving the greater importance of earlier mentions. Our approach captures this mention importance by using attention scores for the mentions in a cluster and combining the mention representations according to their attention scores. We then investigate the effect of the cluster histories by including all the history of the clusters as candidate assignments to the mentions. The resulting system, besides achieving the new SoTA on the CRAC dataset (whether including and excluding non-referring expressions and singletons), achieves CONLL scores equivalent to the current SoTA system not fine-tuned on BERT (Kantor and Globerson, 2019) on CONLL data as well (in which non-referring expressions and singletons are not annotated).

Our third and final contribution is the finding that training our system on annotations of singleton mentions and non-referring expressions enhance its performance on non-singleton coreference chains. By evaluating our system on the CRAC data we show that gains of up to 1.4 percentage points on non-singleton coreference chains can be achieved by training the model with additional singleton mentions and non-referring expressions.

2. System architecture

Anaphora resolution is the task of identifying the referring mentions in a text and assigning those mentions to disjoint clusters such that mentions in the same cluster refer to the same entity. The first subtask of anaphora resolution is mention detection, i.e., extracting candidate mentions from the document. Until recently, most coreference systems selected mentions prior to coreference resolution via heuristic methods often based on parse trees (Björkelund and Kuhn, 2014; Clark and Manning, 2015; Clark and Manning, 2016a; Clark and Manning, 2016b; Wiseman et al., 2015; Wiseman et al., 2016). Lee et al. (2017) introduced a neural network approach for joint mention detection and coreference resolution, obtaining the best performing system at the time. The system was further extended by Lee et al. (2018), Kantor and Globerson (2019), Joshi et al. (2019b) and Joshi et al. (2019a), the current SoTA on the CONLL data set.

[3]The oracle clusters are created from system mention using gold cluster information.

Algorithm 1: Cluster ranking algorithm.

Input: $(\hat{N}_i^*, s_m(i), s_\epsilon(i), \beta(i))_{i=1}^{\lambda T}$
Output: $C_{\lambda T}$

$m = 0; C_0 = \{\}; s_{c_0} = \{\}$;
for $i : 1..\lambda T$ **do**
  TMP $\leftarrow s_\epsilon(i)$;
  for $j : 1..m$ **do**
    TMP $\leftarrow s_m(i) + s_c(j) + s_{mc}(i, j)$
  end
  $b = \arg\max$ TMP;
  if $b == \epsilon$ **then**
    $C_i = C_{i-1} \cup \{\hat{N}_i^*\}$;
    $s_{c_i} = s_{c_{i-1}} \cup s_m(i)$;
    $m = m + 1$;
  else
    $C_i^b = \sum_{m \in C_{i-1}^b \cup \hat{N}_i} a_{i-1}^b(m) \cdot \hat{N}_m^*$;
    $s_{c_i}(b) = \sum_{m \in C_{i-1}^b \cup \hat{N}_i} a_{i-1}^b(m) \cdot s_m(m)$;
  end
end

Our model is also a joint system that predicts mentions and assigns them to the clusters jointly. For a given document D with T tokens, we define all possible spans in D as $N_{i=1}^I$ where $I = \frac{T(T+1)}{2}$, s_i, e_i are the start and the end indices of N_i where $1 \leq i \leq I$. The task for a joint system is to partition all the spans (N) into a sequence of clusters $(C^m)_{m=1}^M$ such that every mention in a specific cluster C^m refers to the same entity. Let C_i be the partially completed clusters up to span N_i. The set of possible assignments for N_i is defined as all the clusters up to the previous span (C_{i-1}) and a special label ϵ. The ϵ is used for three situations: a span is not a mention, or is a non-referring expression, or is the first mention of a cluster.

2.1. Mention Representation

We use a mention representation based on those in (Lee et al., 2018; Kantor and Globerson, 2019). Our system represents a candidate span with the outputs of a BiLSTM, encoding the sentences in a document from both directions to obtain a representation for each token in the sentence. The BiLSTM takes as input the concatenated embeddings $((x_t)_{t=1}^T)$ of both word and character levels. For word embeddings, GloVe (Pennington et al., 2014) and BERT (Devlin et al., 2019) embeddings are used. Character embeddings are learned by a convolution neural networks (CNN) during training. The tokens are represented by concatenated outputs from the forward and the backward LSTMs. The token representations $(x_t^*)_{t=1}^T$ are used together with head representations (h_i^*) to represent candidate spans (N_i^*). The h_i^* of a span is obtained by applying an attention over its token representations ($\{x_{s_i}^*, ..., x_{e_i}^*\}$), where s_i and e_i are the indices of the start and the end of

the span respectively. Formally, we compute h_i^*, N_i^* as follows:

$$\alpha_t = \text{FFNN}_\alpha([x_t^*, \phi(t)])$$
$$a_{i,t} = \frac{exp(\alpha_t)}{\sum_{k=s_i}^{e_i} exp(\alpha_k)}$$
$$h_i^* = \sum_{t=s_i}^{e_i} a_{i,t} \cdot x_t$$
$$N_i^* = [x_{s_i}^*, x_{e_i}^*, h_i^*, \phi(i)]$$

where $\phi(t), \phi(i)$ are the cluster position and span width feature embeddings respectively.

To make the task computationally tractable, our model only considers the spans up to a maximum length of l, i.e. $e_i - s_i < l$, $(s_i, e_i) \in N$. Further pruning is applied before feeding the candidate mentions to the coreference resolver. The top ranked λT spans are selected from lT candidate spans ($\lambda < l$) by a scoring function s_m. where:

$$s_m(i) = \text{FFNN}_m(N_i^*)$$

The top λT selected spans are required *not* to be partially overlap, i.e. there is no such cases that $s_i < s_j \leq e_i < e_j$ or $s_j < s_i \leq e_j < e_i$. The nested spans are not affected by this constrains since they are not partially overlap.

2.2. The Cluster Ranking Model

Let $(\hat{N}_i)_{i=1}^{\lambda T}$ denote the top ranked λT candidate mentions selected by the mention detector after pruning. The model builds the clusters $(C^m)_{m=1}^M$ by visiting $\hat{N}_i$ in text order and assigning them a cluster in the case $i \neq \epsilon$, or creating a new cluster if $i = \epsilon$. Let C_i be the partial clusters consisting of up to i_{th} mentions, and c_i the cluster assigned to $\hat{N}_i$. The task of our cluster ranking model is to output $\hat{C}$ that maximises the score of the final clusters:

$$\hat{C} = \underset{c_1,\ldots,c_{\lambda T}}{\arg\max} \sum_{i=1}^{\lambda T} s(i, c_i)$$

where $s(i,j)$[4] is a scoring function between a mention N_i and a set of possible assignments $j \in \{\epsilon, C_{i-1}^m\}$:

$$s(i,j) = \begin{cases} s_\epsilon(i) & j = \epsilon \\ s_m(i) + s_c(j) + s_{mc}(i,j) & j \neq \epsilon \end{cases}$$

and $s_\epsilon(i)$ is the probability that $\hat{N}_i$ does *not* belongs to any of the previous clusters C_{i-1}^m. To use a scoring function for ϵ instead of a constant 0 (used by Lee et al. (2018)) gives us the flexibility to extend the function for handing more detailed types of ϵ, such as non-referring. $s_m(i)$ is the mention score that has been used to rank the candidate mentions. $s_c(j)$ is the cluster score computed from the mention scores that belongs to the cluster. $s_{mc}(i,j)$ is a pairwise score between i_{th} mention $\hat{N}_i$ and j_{th} partial cluster of C_{i-1}^j. To implement the cluster ranking model we use an attention function $a(m)$ (Bahdanau et al., 2014) to assign an importance to each of the mentions. We compute the cluster score $s_c(j)$ and the cluster representation ($C_{i-1}^{j^*}$) (for computing $s_{mc}(i,j)$), by mention scores/representations and with consideration of mention importance. More precisely, we compute the scores as follows:

$$s_\epsilon(i) = \text{FFNN}_\epsilon(\hat{N}_i^*)$$
$$s_m(i) = \text{FFNN}_m(\hat{N}_i^*)$$
$$\beta(i) = \text{FFNN}_\beta([\hat{N}_i^*, \phi(i_\beta)])$$
$$a_{i-1}^j(m) = \frac{exp(\beta(m))}{\sum_{k \in C_{i-1}^j} exp(\beta(k))}$$
$$s_{c_{i-1}}(j) = \sum_{m \in C_{i-1}^j} a_{i-1}^j(m) \cdot s_m(m)$$
$$C_{i-1}^{j^*} = \sum_{m \in C_{i-1}^j} a_{i-1}^j(m) \cdot \hat{N}_m^*$$
$$F_{(i,j)}^* = [\hat{N}_i^*, C_{i-1}^{j^*}, \hat{N}_i^* \circ C_{i-1}^{j^*}, \phi(i,\hat{j}), \phi(j)]$$
$$s_{mc}(i,j) = \text{FFNN}_{mc}(F_{(i,j)}^*)$$

Both $s_c(j)$ and $C_{i-1}^{j^*}$ are updated each time a cluster is expanded. $\phi(i_\beta)$ is the position embeddings that indicates the position of a mention in the cluster. $\phi(i,\hat{j})$ is a small set of features between the $\hat{N}_i$ and the newest mention $\hat{N}_{\hat{j}}$ of the cluster. We used the same features as Lee et al. (2018): these include genre, speaker (boolean, same or not) and distance (between i and $\hat{j}$) features. $\phi(j)$ is cluster size, a common entity-level feature (Björkelund and Kuhn, 2014). The size is assigned into buckets according to its value. We use the buckets of Björkelund and Kuhn (2014), assigning the values in 8 buckets ([1,2,3,4,5-7,8-11,12-19,20+]). The pseudo-code of our model is shown in Algorithm 1.[5]

2.3. Cluster History

One of the advantages of the mention ranking model is that the correct cluster can be built by attaching the active mention to any of the antecedents in the correct cluster. This reduces the complexity of the task as there are multiple correct links. By contrast, in a standard cluster ranking model, only one correct cluster can be chosen. In order to make multiple links possible in our cluster ranking system, we extended our model by including all cluster histories (CH); this maximises the chance of choosing the correct clusters. (We make sure a mention is always attached to the latest version of the cluster by including an additional pointer linking every cluster history to the latest version of the cluster.) This makes the model slightly more similar to a mention ranking model; however, there is still a fundamental difference, as we use cluster representations instead of mention representations. We replace the line 13 and 14 of Algorithm 1 to get the model that includes cluster histories:

$$b = \text{LATEST}(b)$$
$$C_i = C_{i-1} \cup \sum_{m \in C_{i-1}^b \cup \hat{N}_i} a_{i-1}^b(m) \cdot \hat{N}_m^*$$

[4] We follow Lee et al. (2018) and use i to indicate the anaphor and j for the antecedent.

[5] We do *not* use coarse-to-fine pruning or higher-order inference, unlike Lee et al. (2018) and Kantor and Globerson (2019). We found coarse-to-fine pruning does *not* improve our model when compared with simpler distance pruning. As for higher-order inference, our system already has access to the entity-level information by default, hence it is not necessary.

$$s_{c_i} = s_{c_{i-1}} \cup \sum_{m \in C^b_{i-1} \cup \hat{N}_i} a^b_{i-1}(m) \cdot s_m(m)$$
$$m = m + 1$$

where LATEST(b) is a function to find the latest version of the cluster b.

2.4. Identifying Non-Referring Expressions

To add non-referring expressions identification, we extend ϵ into multiple classes: NO for non-mention, NR for non-referring and DN for discourse new, including singletons

$$s_\epsilon(i) = \begin{cases} s_{no}(i) & \text{NO} \\ s_{nr}(i) + s_m(i) & \text{NR} \\ s_{dn}(i) + s_m(i) & \text{DN} \end{cases}$$

Several non-referring types are annotated in the ARRAU corpus: in addition to expletives, there are also predicative NPs (e.g., *a policeman* in *John is a policeman*), non-referring quantifiers (e.g.,*nobody* in *I see nobody here*) (Karttunen, 1976), idioms (e.g., *her hand* in *He asked her for her hand*), etc. As we will see, the basic NR classifier can be extended to do a fine-grained classification of non-referring expressions.

By distinguishing 'non-mentionhood' from non-anaphoricity the system naturally resolves singletons (i.e. the clusters with a size of one). Non-referring expressions are usually filtered before building the coreference chains, e.g. in MARS (Mitkov et al., 2002); we will call this PREFILTERING approach. In the PREFILTERING approach, the system removes the markables identified as non-referring expressions from further processing once they have been identified. To be more specific, we replace line 8 of algorithm 1 with:

if $b ==$ NO **or** $b ==$ NR **then**
  $C_i = C_{i-1}$; $s_{c_i} = s_{c_{i-1}}$; $m = m$;
else if $b ==$ DN **then**

The PREFILTERING approach is aggressive, which might have a negative effect on results if referring expressions have been filtered incorrectly. We also tried therefore a second approach: only do prefiltering when the non-referring expressions classifier has a softmax score above a heuristic threshold t ($0 \leq t \leq 1$)). The softmax score is calculated between previous clusters and classes in ϵ (i.e. TMP in algorithm 1). If the score is below this threshold, non-referring expressions are identified after (postfiltering) forming the clusters (we call this HYBRID approach). During postfiltering, candidates that are classified as non-referring markables with lower confidence and are not part of clusters are included as additional non-referring markables.

2.5. Learning

To train a cluster ranking model on system clusters is challenging, as we need to find a way to learn from the partially correct clusters. It is also slow, as the system processes one mention at a time, hence cannot benefit largely from parallel computing. The solution we adopted was training the model on oracle clusters. This is simpler and faster, since the clusters for one training document can be created before computing more heavy stuff, e.g. the cluster scores $s_c(j)$ and pairwise scores $s_{mc}(i, j)$. More precisely, we create the oracle clusters during the training using gold cluster ids; system mentions belonging to the same gold clusters are grouped. This is much faster than training the model on the system mentions directly, since training on the system mentions requires computing scores for each mention separately. In a preliminary experiment, we discovered that by training on oracle clusters we obtain not only a better CONLL score, but also a fivefold speedup compared with the model trained on the system mentions directly.[6]

As a loss function, we optimize on the marginal log-likelihood of all the clusters that contain mentions from the same gold cluster GOLD(i) of $\hat{N}_i$. Formally,

$$log \prod_{i=1}^{\hat{N}} \sum_{\hat{c} \in C_{i-1} \cap \text{GOLD}(i)} P(\hat{c})$$

In case C_{i-1} does not contain any mention from GOLD(i) or $\hat{N}_i$ does not belongs to a gold cluster, we set GOLD$(i) = \{\epsilon\}$. For our model to have more than one class in ϵ, the GOLD(i) is set to the relevant classes (NO,NR or DN).

Parameter	Value
BiLSTM layers/size/dropout	3/200/0.4
FFNN layers/size/dropout	2/150/0.2
CNN filter widths/size	[3,4,5]/50
Char/GloVe/Feature embedding size	8/300/20
BERT embedding size/layer	1024/Last 4
Embedding dropout	0.5
Max span width (l)	30
Max num of clusters	250
Mention/token ratio (λ)	0.4
Optimiser	Adam (1e-3)
Training step	200K

Table 1: Hyperparameters for our models.

3. Data and Hyperparameters

For full anaphora resolution, our primary evaluation dataset was the CRAC 2018 corpus (Poesio et al., 2018). In addition, we evaluated our model on the PD corpus, also containing expletives. Finally, we evaluated our model on the CONLL 2012 English corpora (Pradhan et al., 2012) to compare its performance with the SoTA on the CONLL task.

The CRAC Task 1 dataset is based on the RST portion of the ARRAU corpus (Uryupina et al., 2019). The annotation scheme specifies the annotation of referring expressions (including singletons) and non-referring expressions; split antecedent plurals, generic references, and discourse deixis are annotated, as well as bridging references. The RST portion of ARRAU consists of news texts (1/3 of the PENN Treebank), with 228,000 tokens and 72,000 mentions.

PD is a constantly growing corpus collected using the annotation game Phrase Detectives (Poesio et al., 2019). The corpus was annotated by players and then aggregated by

[6] We train both approaches on the CONLL data for 200K steps on a GTX 1080Ti GPU. It takes 16 and 80 hours to train a model on oracle and system mentions respectively.

	Models	MUC			B^3			$CEAF_{\phi_4}$			Avg. F1
		P	R	F1	P	R	F1	P	R	F1	
Singletons included	PREFILTERING	75.5	**79.0**	77.2	75.9	**80.7**	78.2	75.2	77.3	76.2	77.2
	HYBRID	**77.9**	78.5	**78.2**	**77.4**	80.3	**78.8**	**75.4**	78.1	**76.8**	**77.9**
	FINE NR	76.7	77.3	77.0	76.8	79.7	78.2	74.9	78.0	76.4	77.2
	Lee et al. (2013)*	72.1	58.9	64.8	77.5	77.1	77.3	64.2	**88.1**	74.3	72.1
Singletons excluded	PREFILTERING	75.5	**79.0**	77.2	67.0	**73.0**	69.9	67.1	**65.1**	66.1	71.1
	HYBRID	**77.9**	78.5	**78.2**	**69.2**	71.8	**70.4**	**69.5**	63.8	**66.5**	**71.7**
	FINE NR	76.7	77.3	77.0	68.0	70.7	69.3	66.6	64.2	65.4	70.6
	NO NR	76.7	77.0	76.8	68.7	69.7	69.2	66.1	63.8	64.9	70.3
	Lee et al. (2013)*	72.3	58.9	64.9	67.9	48.5	56.5	54.2	53.0	53.6	58.3

Table 2: The comparison between our models and the SoTA system on the CRAC test set. * indicates systems evaluated on the gold mentions.

an aggregating method to create a silver standard corpus. Both singletons and non-referring markables are annotated. We used the latest release of the corpus, consisting of 542 documents, 408,000 tokens and 108,000 mentions.[7].

The CONLL datasets are the standard datasets for coreference. The English CONLL corpus consists of 3493 documents for a total of 1.6M tokens and 194,000 mentions.

We use the official CONLL 2012 scorer to score our predictions when evaluating without singletons and non-referring markables, and the official CRAC 2018 scorer (Poesio et al., 2018) to evaluate other cases. The CRAC 2018 Extended Scorer is an extension of the CONLL 2012 official scorer developed by Nafise Moosavi that can handle singletons and non-referring markables. The Extended Scorer is identical to the CONLL scorer when evaluating without singletons and non-referring markables, but also reports P, R and F1 values for non-referring markables when those are considered. Following standard practice, we report recall, precision, and F1 scores for MUC, B^3 and $CEAF_{\phi_4}$ and the average F1 score of those three metrics. Besides, we report the F1 score for non-referring when needed.

For our experiments, we use the same maximum span width ($l = 30$), number spans per tokens ($\lambda = 0.4$) and most of the network parameters as Lee et al. (2018) and Kantor and Globerson (2019). The details are in Table 1.

4. Results and Discussions

4.1. Evaluation on the CRAC data set

We first compared the two proposed approaches for using non-referring expressions, PREFILTERING and HYBRID. For our HYBRID model, we set the threshold (t) to 0.5 after tuning on the development set. Table 2 shows the results of our models on the CRAC test set. As expected, the HYBRID model, using a less greedy pruning, achieved better F1 scores on all three coreference metrics. In terms of the non-referring scores (see Table 3), the PREFILTERING approach has better recall and F1 score, while the HYBRID approach has better precision. We hypothesize this is mainly because the PREFILTERING approach generates more non-referring expressions due to its greedy pruning– i.e., the PREFILTERING approach keeps all the candidate non-referring markables once they are identified–while the HYBRID approach favours the coreference clusters for non-referring markables fall below the threshold. The HYBRID approach has a better overall performance according to our weighed F1 scores (0.85 * COREF F1 + 0.15 * NR F1) The weights are determined by the proportion of the referring and non-referring markables in the corpus.

[7]https://github.com/dali-ambiguity/Phrase-Detectives-Corpus-2.1.4

Models	P	R	F1
PREFILTERING	76.6	74.5	75.5
HYBRID	78.0	72.4	75.1
FINE NR	77.0	75.5	76.3

Table 3: The scores for non-referring expressions of our models on the CRAC test set.

NR types	P	R	F1
Expletive	93.8	100.0	96.8
Predicate	77.6	75.2	76.4
Quantifier	65.0	64.7	64.9
Coordination	77.5	82.0	79.7
Idiom	77.0	55.9	64.8

Table 4: The scores of our models on the fine-grained non-referring types.

Fine-grained Non-referring

We further extended the basic NR classifier to recognise the more fine-grained classification of non-referring expressions annotated in the CRAC dataset by configuring our HYBRID model to learn from the fine-grained types (FINE NR). Our model does very well on resolving expletives (96.8% F1) and achieves 76 - 80% F1 score on predicates and coordinations, but has a lower F1 score of around 65% on recognising non-referring quantifiers and idioms. We also compared this model with the other models to dealing with non-referring expressions by collapsing the classifications it produces (Table 3). As we can see from that Table, although the task is harder, using the fine-grained types for training results in slightly better performance on identifying non-referring markables in general than models trained

	Models	MUC			B^3			$CEAF_{\phi_4}$			Avg. F1
		P	R	F1	P	R	F1	P	R	F1	
Singletons included	Poesio et al. (2019)	79.3	72.5	75.7	72.1	69.3	70.7	70.5	73.2	71.8	72.7
	Our model	**81.9**	**76.4**	**79.1**	**74.9**	**73.7**	**74.3**	**72.2**	**75.1**	**73.6**	**75.7**
Singletons excluded	Poesio et al. (2019)	79.3	72.5	75.7	58.3	52.4	55.2	58.3	49.5	53.5	61.5
	Our model	**81.9**	**76.4**	**79.1**	**64.7**	**61.0**	**62.8**	**62.9**	**54.8**	**58.6**	**66.8**

Table 5: The comparison between our models and the SoTA system on the PD test set.

	Models	MUC			B^3			$CEAF_{\phi_4}$			Avg. F1
		P	R	F1	P	R	F1	P	R	F1	
Context Independent Embeddings	Clark and Manning (2016a)	79.2	70.4	74.6	**69.9**	58.0	63.4	63.5	55.5	59.2	65.7
	Lee et al. (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	**59.0**	60.8	67.2
	Zhang et al. (2018)	**79.4**	**73.8**	**76.5**	69.0	**62.3**	**65.5**	**64.9**	58.3	**61.4**	**67.8**
Pre-trained Contextual Embeddings	Lee et al. (2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
	Kantor and Globerson (2019)	82.6	**84.1**	**83.4**	73.3	**76.2**	**74.7**	**72.4**	**71.1**	**71.8**	**76.6**
	Our model	**82.7**	83.3	83.0	**73.8**	75.6	**74.7**	72.2	71.0	71.6	76.4
Fine-tuned on BERT	Joshi et al. (2019b)	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
	Joshi et al. (2019a)	**85.8**	**84.8**	**85.3**	**78.3**	**77.9**	**78.1**	**76.4**	**74.2**	**75.3**	**79.6**

Table 6: Comparison between our models and the top performing systems on the CONLL test set.

on a single NR class. In term of the performance on coreference chains, the FINE NR approach achieved the same score as the PREFILTERING approach and slightly lower than the HYBRID approach (see Table 2).

Training without Singletons and Non-referring Finally, we trained our model without singletons and non-referring expressions (NO NR) to assess their effects on non-singleton clusters (i.e. the standard CONLL setting). Since here we evaluate in a singleton excluded setting, we report for our models trained with singletons and non-referring expressions the standard CONLL scores with singletons and non-referring markables excluded. As shown in Table 2, all three models trained with additional singleton and non-referring markables achieved better CONLL scores when compared with the newly trained model. The system achieves substantial gains of up to 1.4 percentage points (HYBRID) by training with the additional singletons and non-referring expressions. This suggests that the availability of singletons and non-referring markables can help the decisions made for non-singleton clusters.

State-of-the-art Comparison Since the CRAC corpus was released recently, the only published results are those by the baseline system (Lee et al., 2013) on the shared task (Poesio et al., 2018). Our best system (HYBRID) outperforms this baseline by large margins (5.8% and 13.4% when evaluated with or without singletons respectively) (see Table 2) even though that system was evaluated on gold mentions.

4.2. Evaluation on the PD data set

We then test our best system on the PD corpus[8]. We compare our system with the results by Poesio et al. (2019) (Table 5). Our system is 3% better when evaluated with singletons included and outperforms their system by 5.3% when evaluated without the singletons. In addition, our system achieved an F1 of 56.7% on non-referring expressions and this is 2.1% better than their result (54.6%). Overall, our system achieved the new SoTA on the PD data.

4.3. Evaluation on the CONLL data set

Finally, we tested our models on the CONLL data to assess the performance of our system on the standard data set. Table 6 compares our results with those of the top-performing systems on CONLL at the present time. We report precision, recall and F1 scores for all three major metrics (MUC, B^3 and $CEAF_{\phi_4}$) and mainly focus on the average CONLL F1 scores presented in the last column. As showed in Table 6, our model achieved a CONLL score of 76.4%, which is only 0.2% lower than the best-reported result at present, achieved by (Kantor and Globerson, 2019) that use a similar mention representations as our system. Although the systems by Joshi et al. (2019b) and Joshi et al. (2019a) have better results than the Kantor and Globerson (2019) system, it is not directly comparable with our system, as their systems are fine-tuned on BERT. Such systems need to be trained on GPUs with 32GB memory, which are not available to our group. By contrast, our system was trained with a GTX 1080Ti GPU with an 11GB memory.

4.4. Discussion

We further analyze our model on the CONLL data to give a more detailed study on different aspects of our model. (We use the standard CONLL data instead of the CRAC data because the CONLL corpus is larger than the CRAC corpus and is widely used. As a result, the analysis on CONLL data might also be beneficial for other researchers focusing on CONLL only.)

Mention Importance We first assess our hypothesis that our attention scores can capture mention importance–i.e., the finding from the linguistic and psychological literature on anaphora that the initial mentions of an entity tend to

[8]Poesio et al. (2019) uses an early version of our system.

	Positions				
Size	1	2	3	4	5-7
2	0.55	0.45			
3	0.38	0.32	0.29		
4	0.29	0.24	0.23	0.22	
5	0.24	0.20	0.19	0.19	0.19
6	0.19	0.17	0.16	0.17	0.15
7	0.18	0.14	0.14	0.14	0.13

Table 7: The average mention importance attention scores in the CONLL development set, grouped by mentions position and cluster size in the final clusters.

	Avg. F1	Δ
Our model	76.9	
- Position emb	76.2	0.7
- Width emb	76.5	0.4
- Cluster history	75.9	1.0
- Oracle cluster	76.3	0.6

Table 8: The comparison between our best model and different ablated models on CONLL development set.

include more information, whereas the following mentions are generally reduced. Table 7 shows an analysis of the attention scores that supports this hypothesis. We computed the average attention scores for mentions in a cluster in order of mention. Clusters that have different size are analysed separately, as scores from different-sized clusters are not directly comparable. As we can see from the Table, after analysis the attention scores assigned to the mentions at different positions in the cluster, we find that the attention scores assigned to the first mention in a cluster are always higher than others, which is in line with linguistic findings that mentions introducing an entity are more informative. This suggests that our attention model does capture something like mention importance.

Why Cluster Ranking? The reason why we use a cluster ranking approach instead of mention ranking is not only because it is linguistically more appealing, but also due to several practical restrictions of the mention ranking models. First of all, the current SoTA mention-ranking systems tend to be hybrids, using entity-level features alongside mention-pair features. Thus, such models are usually more complex than pure mention ranking models, and substantially increase the number of trainable parameters. Take Lee et al. (2018) system as an example. The mention ranking part of the system contains 4.8M parameters, but the full system has double the number of parameters (9.6M) to access entity-level features. Our system, on the other hand, links the mentions directly to the entity and uses only 4.8M parameters, which is much simpler than such hybrid models. Second, we hope that using a cluster ranking model will allow us to explore rich cluster level features and advanced search algorithms (e.g. beam search) in future work.

The Effect of Oracle Clusters on Training Time Training cluster ranking systems using system clusters is time-consuming: Our model trained on system clusters takes 80 hours to train for 200K steps, which is much more than the 48 hours training time of the Lee et al. (2018) system (400K steps). The main reason the cluster ranking system is slower than its mention ranking counterpart is that the cluster ranking model processes one mention at a time, hence does not benefit from parallelization. To solve this problem, we trained the system on oracle clusters instead. The oracle clusters are created by using the system mentions with the gold cluster ids. By doing so all the clusters can be created before resolving the mentions into the entities. As a result, the training (200K steps) can be finished in as little as 16 hours, which is 5x faster than training the model on system clusters, and 3x faster than training the mention ranking model.

4.5. Ablation study

We removed different parts of our model to show the importance of the individual part of our system (see Table 8).

Position Embeddings We first removed the position embeddings, used in the self-attention to determine the relative importance of the mentions in the cluster. By removing the position embeddings, the relative importance of a mention becomes independent of its position in the cluster. As a result, the performance of the model drops by 0.7%.

Width Embeddings We then removed the cluster width embeddings from our features. The cluster width embedding is a feature used in computing the pairwise scores, which allows mentions to known the size of individual candidate clusters. (Cluster size can be used as an indicator of cluster salience, as the larger the size, the more frequently an entity is mentioned, having therefore a higher salience.) The cluster width feature contributes 0.4% towards our model.

Cluster History We trained a model that keeps exactly one cluster per entity, and the history clusters are excluded from the candidate lists. This removing of history clusters reduces the chance of linking the mentions to the correct entity; as a consequence, the performance drops by 1 percentage point.

Oracle Clusters Finally, we trained a model using the system clusters directly instead of the oracle clusters. As we mentioned in the previous section, training on the system clusters is more time consuming than training on the oracle clusters. And replacing these clusters suggests that training on the oracle clusters is not only faster, but also results in better performance (0.6%).

5. Related Work

Pure Mention Ranking Models Most recent coreference systems are highly reliant on mention ranking, which is effective and generally faster to train compared with the cluster ranking system. Systems based only on the mention ranking model include (Wiseman et al., 2015; Clark and Manning, 2016b; Lee et al., 2017). Wiseman et al. (2015) introduced a neural network based approach to solve the task in a non-linear way. In their system, the heuristic features commonly used in linear models are transformed by a $\tanh$ function to be used as the mention representations. Clark and Manning (2016b) integrated reinforcement learning to let the model optimize directly on the B^3 scores. Lee

et al. (2017) first presented a neural joint approach for mention detection and coreference resolution. Their model does not rely on parse trees; instead, the system learns to detect mentions by exploring the outputs of a BiLSTM.

Models using Entity Level Features Researchers have been aware of the importance of entity level information at least since Luo et al. (2004), and many systems trying to exploit cluster based features have been proposed since. Among neural network models, Björkelund and Kuhn (2014) built a latent tree system that explores nonlocal features through beam search. The global feature-aided model showed clear gains when compared with the model based only on pairwise features. Clark and Manning (2015) introduced a entity-centric coreference system by manipulating the scores of a mention pair model. The system first runs a mention pair model on the document and then uses an agglomerative clustering algorithm to build the clusters in an easy-first fashion. This system was later extended by Clark and Manning (2016b) to make it run on neural networks. Wiseman et al. (2016) add to the Wiseman et al. (2015) system an LSTM to encode the partial clusters. The outputs of the LSTM are used as additional features for the mention ranking model. Lee et al. (2018) is an extended version of Lee et al. (2017) mainly enhanced by using ELMo embeddings (Peters et al., 2018), but the use of second-order inference enabled the system explore partial entity level features and further improved the system by 0.4 percentage points. Later the model was further improved by Kantor and Globerson (2019) who use BERT embeddings (Devlin et al., 2019) instead of ELMo embeddings. At this stage, both BERT and ELMo embeddings are used in a pre-trained fashion. Recently, Joshi et al. (2019b) fine-tunes the BERT model for coreference task, result in again a small improvement. Later, Joshi et al. (2019a) introduces a BERT model (SpanBERT) specifically trained for the tasks that involves spans, by using the SpanBERT, the system achieved a substantial gain of 2.7% when compared with the Joshi et al. (2019b) model.

Cluster Ranking Models To the best of our knowledge, our system is the only recent system that does *not* rely on a mention ranking model. However, there are a number of early studies that laid a solid foundation for the cluster ranking models (see (Poesio et al., 2016a) for a survey). The best known 'modern' examples are the systems proposed by Luo et al. (2004) and by Rahman and Ng (2011), but this approach was the dominant model for anaphora resolution at least until the paper by Soon et al. (2001), as it directly implements the linguistically and psychologically motivated view that anaphora resolution involves the creation of a discourse model articulated around discourse entities (Karttunen, 1976). The entity mention model of Luo et al. (2004) introduced the notion that a training instance consists of a mention and an active cluster, and therefore allowed for cluster-level features encoding information about multiple entities in the cluster. Luo et al. (2004) also proposed a clustering algorithm in which the clustering options are encoded in a Bell tree that also specifies the coreference decisions resulting in a cluster–an idea related to our idea of cluster history. Rahman and Ng (2011) introduced the term 'cluster ranking' and greatly developed the approach, e.g., by introducing a rich set of cluster-level features. Their model was the first cluster-ranking model to significantly outperform mention pair models.

Singletons and Non-referring Expressions Again, to the best of our knowledge, ours is the only modern neural network-based, full coreference system that attempts to output singletons and non-referring markables. The Stanford Deterministic Coreference Resolver (Lee et al., 2013) uses a number of filters to *exclude* expletives as well as quasi-referring mentions such as percentages (e.g., *9%*) and measure NPs (e.g., *a liter of milk*) and its extension proposed by De Marneffe et al. (2015) includes more fiters to exclude singletons, but these aspects of the system are not evaluated. The best-known systems also attempting to annotate non-referring markables date back to the pre-ONTONOTES era. The pronoun resolution algorithm proposed by Lappin and Leass (1994) includes a series of hand-crafted heuristics to detect expletives. The statistical classifier proposed by Evans (2001) classifies pronouns in several categories which, apart from nominal anaphoric, include cataphoric, pleonastic, and clause-anaphoric. Versley et al. (2008) used the BBN pronoun corpus to confirm the hypothesis that tree kernels would be well-suited to identify expletive pronouns. Boyd et al. (2005) develop a set of hand-crafted heuristics to identify non-referring *nominals* in the sense of Karttunen (1976). The systems developed by Bergsma and colleagues to identify pronominal *it* with a classifier using a combination of lexical features and web counts (Bergsma et al., 2008; Bergsma and Yarowsky, 2011). A lot of work on identifying expletives was carried out in the context of the DiscoMT evaluation campaigns, but this work was typically only focused on disambiguating pronoun *it* (Loáiciga et al., 2017). For more discussion of these and other systems, see (Uryupina et al., 2016).

6. Conclusions

In this work, we presented the first neural network based system for full coreference resolution also covering singletons and non-referring markables. Our system uses an attention mechanism to form the cluster representations using mention importance scores from the mentions belonging to the cluster. By training the system on oracle clusters we show that a cluster ranking system can be trained 5x faster, and faster than a mention-ranking system with a similar architecture. Evaluation on the CRAC corpus shows that our system is 5.8% better than the only existing comparable system, the Shared Task baseline system that used the gold mentions. The evaluation on PD shows the same trend. Further evaluation on the CONLL corpus shows our system achieves on that corpus, for the subtask in which singleton and non-referring expression detection are excluded, a performance equivalent to that of the SoTA Kantor and Globerson (2019) system. We also demonstrated that a large improvement on non-singleton coreference chains can be made by training the system with additional singletons and non-referring expressions.

7. Acknowledgments

This research was supported in part by the DALI project, ERC Grant 695662.

8. Bibliographical References

Ariel, M. (1990). *Accessing Noun-Phrase Antecedents*. Croom Helm Linguistics Series. Routledge.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bergsma, S. and Yarowsky, D. (2011). Nada: A robust system for non-referential pronoun detection. In Iris Hendrickx, et al., editors, *Anaphora Processing and Applications*, pages 12–23, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bergsma, S., Lin, D., and Goebel, R. (2008). Distributional identification of non-referential pronouns. In *Proceedings of ACL-08: HLT*, pages 10–18, Columbus, Ohio, June. Association for Computational Linguistics.

Björkelund, A. and Kuhn, J. (2014). Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 47–57.

Boyd, A., Gegg-Harrison, W., and Byron, D. (2005). Identifying non-referential it: a machine learning approach incorporating linguistically motivated patterns. In *In Proceedings of the ACL Workshop on Feature Selection for Machine Learning in NLP*, pages 40–47, Ann Arbor.

Chen, H., Fan, Z., Lu, H., Yuille, A., and Rong, S. (2018). Preco: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium, October-November. Association for Computational Linguistics.

Clark, K. and Manning, C. D. (2015). Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*.

Clark, K. and Manning, C. D. (2016a). Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Clark, K. and Manning, C. D. (2016b). Improving coreference resolution by learning entity-level distributed representations. In *Association for Computational Linguistics (ACL)*.

De Marneffe, M.-C., Recasens, M., and Potts, C. (2015). Modeling the lifespan of discourse entities with application to coreference resolution. *J. Artif. Int. Res.*, 52(1):445–475, January.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Evans, R. (2001). Applying machine learning toward an automatic classification of it. *Literary and linguistic computing*, 16(1):45–58.

Fernandes, E. R., dos Santos, C. N., and Milidiú, R. L. (2014). Latent trees for coreference resolution. *Computational Linguistics*, 40(4):801–835, December.

Guillou, L. and Hardmeier, C. (2016). Protest: A test suite for evaluating pronouns in machine translation. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Hardmeier, C., Nakov, P., Stymne, S., Tiedemann, J., Versley, Y., and Cettolo, M. (2015). Pronoun-focused mt and cross-lingual pronoun prediction: Findings of the 2015 discomt shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, September. Association for Computational Linguistics.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2019a). Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019b). 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. Association for Computational Linguistics.

Kantor, B. and Globerson, A. (2019). Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy, July. Association for Computational Linguistics.

Karttunen, L. (1976). Discourse referents. In *Syntax and Semantics 7 - Notes from the Linguistic Underground*. Academic Press.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–562.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Lee, K., He, L., and Zettlemoyer, L. S. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Loáiciga, S., Guillou, L., and Hardmeier, C. (2017). What is it? disambiguating the different readings of the pronoun 'it'. In *Proc. of EMNLP*.

Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., and Roukos, S. (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*.

Martschat, S. and Strube, M. (2015). Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

Mitkov, R., Evans, R., and Orasan, C. (2002). A new, fully automatic version of Mitkov's knowledge-poor pronoun

resolution method. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 168–186. Springer.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. S. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Poesio, M., Stuckardt, R., Versley, Y., and Vieira, R. (2016a). Early approaches to anaphora resolution: Theoretically inspired and heuristic-based. In M. Poesio, et al., editors, *Anaphora Resolution: Algorithms, Resources and Applications*, chapter 3. Springer.

Poesio, M., Stuckardt, R., and Versley, Y. (2016b). *Anaphora Resolution: Algorithms, Resources and Applications*. Springer, Berlin.

Poesio, M., Grishina, Y., Kolhatkar, V., Moosavi, N., Roesiger, I., Roussel, A., Simonjetz, F., Uma, A., Uryupina, O., Yu, J., and Zinsmeister, H. (2018). Anaphora resolution with the arrau corpus. In *Proc. of the NAACL Worskhop on Computational Models of Reference, Anaphora and Coreference (CRAC)*, pages 11–22, New Orleans, June.

Poesio, M., Chamberlain, J., Paun, S., Yu, J., Uma, A., and Kruschwitz, U. (2019). A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation. 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 1778–1789, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

Rahman, A. and Ng, V. (2011). Narrowing the modeling gap: a cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research*, 40:469–521.

Soon, W. M., Lim, D. C. Y., and Ng, H. T. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), December.

Steinberger, J., Poesio, M., Kabadjov, M., and Jezek, K. (2007). Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680. Special issue on Summarization.

Steinberger, J., Kabadjov, M., and Poesio, M. (2016). Coreference applications to summarization. In *Anaphora Resolution: Algorithms, Resources and Applications*, chapter 15. Springer.

Taulé, M., Martí, M. A., and Recasens, M. (2008). Ancora: Multilevel annotated corpora for catalan and spanish. In *LREC 2008*.

Telljohann, H., Hinrichs, E. W., Kübler, S., Zinsmeister, H., and Beck, K.). Stylebook for the tübingen treebank of written german (tüba-d/z).

Uryupina, O., Kabadjov, M., and Poesio, M. (2016). Detecting non-reference and non-anaphoricity. In *Anaphora Resolution: Algorithms, Resources, and Applications*, pages 369–392. Springer, Berlin.

Uryupina, O., Artstein, R., Bristot, A., Cavicchio, F., Delogu, F., Rodriguez, K. J., and Poesio, M. (2019). Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU corpus. *Journal of Natural Language Engineering*.

Versley, Y., Moschitti, A., Poesio, M., and Yang, X. (2008). Coreference systems based on kernels methods. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 961–968, Manchester, UK, August. Coling 2008 Organizing Committee.

Wiseman, S., Rush, A. M., Shieber, S., and Weston, J. (2015). Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1416–1426.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004.

Zhang, R., Nogueira dos Santos, C., Yasunaga, M., Xiang, B., and Radev, D. (2018). Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107. Association for Computational Linguistics.