Neural Coreference Resolution for Arabic

Abdulrahman Aloraini[1,2]* **Juntao Yu**[1]* **Massimo Poesio**[1]
[1]Queen Mary University of London, United Kingdom
[2]Qassim University, Saudi Arabia
{a.aloraini, juntao.yu, m.poesio}@qmul.ac.uk

Abstract

No neural coreference resolver for Arabic exists, in fact we are not aware of any learning-based coreference resolver for Arabic since (Björkelund and Kuhn, 2014). In this paper, we introduce a coreference resolution system for Arabic based on Lee et al's end-to-end architecture combined with the Arabic version of BERT and an external mention detector. As far as we know, this is the first neural coreference resolution system aimed specifically to Arabic, and it substantially outperforms the existing state-of-the-art on OntoNotes 5.0 with a gain of 15.2 points CONLL F1. We also discuss the current limitations of the task for Arabic and possible approaches that can tackle these challenges.

1 Introduction

Coreference resolution is the task of grouping mentions in a text that refer to the same real-world entity into clusters (Poesio et al., 2016) . Coreference resolution is a difficult task that requires reasoning, context understanding, and background knowledge of real-world entities, and has driven research in both natural language processing and machine learning, particularly since the release of the ONTONOTES multilingual corpus providing annotated coreference data for Arabic, Chinese and English and used for the 2011 and 2012 CONLL shared tasks (Pradhan et al., 2012). Since then, there has been substantial research on English coreference, most recently using neural coreference approaches (Lee et al., 2017; Lee et al., 2018; Kantor and Globerson, 2019a; Joshi et al., 2019b; Joshi et al., 2019a; Yu et al., 2020b; Wu et al., 2020), leading to a significant increase in the performance of coreference resolvers for English. By contrast, there has been almost no research on Arabic coreference; the performance for Arabic coreference resolution has not improved much since the CONLL 2012 shared task, and in particular no neural architectures have been proposed–the current state-of-the-art system remains the model proposed in (Björkelund and Kuhn, 2014). In this paper we close this very obvious gap by proposing what to our knowledge is the first neural coreference resolver for Arabic.[1]

One explanation for this lack of research might simply be the lack of training data large enough for the task. Another explanation might be that Arabic is more problematic than English because of its rich morphology, its many dialects, and/or its high degree of ambiguity. We explore the first of these possibilities. Coreference resolution can be further divided into two subtasks–mention detection and mention clustering–as illustrated in Figure 1. In early work, coreference's two subtasks were usually carried out in a pipeline fashion (Soon et al., 2001; Fernandes et al., 2014; Björkelund and Kuhn, 2014; Wiseman et al., 2015; Wiseman et al., 2016; Clark and Manning, 2016a; Clark and Manning, 2016b), with candidate mentions selected prior the mention clustering step. Since Lee et al. (2017) introduced an end-to-end neural coreference architecture that achieved state of the art by carrying out the two tasks jointly, as first proposed by Daume and Marcu (2005), most state-of-the-art systems have followed this approach. However, no end-to-end solution was attempted for Arabic. We intend to explore whether an end-to-end solution would be practicable with a corpus of more limited size.

* Equal contribution. Listed by alphabetical order.

[1] The code is available at https://github.com/juntaoy/aracoref

Proceedings of the 3rd Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2020), pages 99–110,
Barcelona, Spain (online), December 12, 2020.

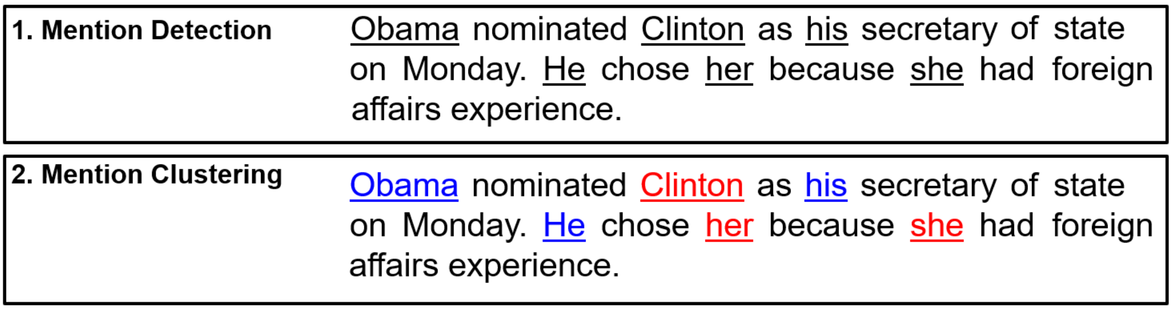
1. Mention Detection
Obama nominated Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience.
2. Mention Clustering
Obama nominated Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience.

Figure 1: The first step in coreference resolution is mention detection. The detected mentions are underlined. The second step is mention clustering. We have two clusters {Obama, his, he} and {Clinton, her, she}. The mention detector might identify other words as mentions, but for simplicity we present only the mentions of the two clusters.

The approach we followed to adapt the state-of-the-art English coreference resolution architecture to Arabic is as follows. We started with a strong baseline system (Lee et al., 2018; Kantor and Globerson, 2019a), enhanced with contextual BERT embeddings (Devlin et al., 2019). We then explored three methods for improving the model's performance for Arabic. The first method is to pre-process Arabic words with heuristic rules. We follow Althobaiti et al. (2014) to normalize the letters with different forms, and removing all the diacritics. This results in a substantial improvement of 7 percentage points over our baseline. The second route is to replace multilingual BERT with a BERT model trained only on the Arabic texts (AraBERT) (Antoun et al., 2020). Multilingual BERT is trained with 100+ languages; as a result, it is not optimized for any of them. As shown by Antoun et al. (2020), monolingual BERT trained only on the Arabic texts has better performance on various NLP tasks. We found the same holds for coreference: using embeddings from monolingual BERT, the model further improved the CONLL F1 by 4.8 percentage points. Our third step is to leverage the end-to-end system with a separately trained mention detector (Yu et al., 2020a). We show that a better mention detection performance can be achieved by using a separately trained mention detector. And by using a hybrid training strategy between the end-to-end and pipeline approaches (end-to-end annealing) our system gains an additional 0.8 percentage points. Our final system achieved a CONLL F1 score of 63.9%, which is is 15% more than the previous state-of-the-art system (Björkelund and Kuhn, 2014) on Arabic coreference with the CONLL dataset. Overall, we show that the state-of-the-art English coreference model can be adapted to Arabic coreference leading to a substantial improvement in performance when compared to previous feature-based systems.

2 Related Work

2.1 English Coreference Resolution

Like with other natural language processing tasks, most state-of-the-art coreference resolution systems are evaluated on English data. Coreference resolution for English is an active area of research. Until the appearance of neural systems, state-of-the-art systems for English coreference resolution were either rule-based (Lee et al., 2011) or feature-based (Soon et al., 2001; Björkelund and Nugues, 2011; Fernandes et al., 2014; Björkelund and Kuhn, 2014; Clark and Manning, 2015). Wiseman et al. (2015) introduced a neural network-based approach to solving 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 bi-directional LSTM. Lee et al. (2018) is an extended version of Lee et al. (2017) mainly enhanced by using ELMo embeddings (Peters et al., 2018), in addition, the use of second-order inference enabled the system to 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

(2019a) who use BERT embeddings (Devlin et al., 2019) instead of ELMo embeddings. In these systems, both BERT and ELMo embeddings are used in a pre-trained fashion. More recently, Joshi et al. (2019b) fine-tuned the BERT model for coreference, resulting in a small further improvement. Later, Joshi et al. (2019a) introduces SPANBERT which is trained for tasks that involve spans. Using SPANBERT, they achieved a substantial gain of 2.7% when compared with the Joshi et al. (2019b) model. Wu et al. (2020) reformulate the coreference resolution task as question answering task and achieved the state-of-the-art results by pretrain the system first on the large question answering corpora.

2.2 Arabic Coreference Resolution

There have been several studies of Arabic coreference resolution; in particular, several of the systems involved in the CONLL 2012 shared task attempted Arabic as well. li (2012) used syntactic parse trees to detect mentions, and compared pairs of mention based on their semantic and syntactic features. Zhekova and Kübler (2010) proposed a language independent module that requires only syntactic information and clusters mentions using the memory-based learner TiMBL (Daelemans et al., 2004). Chen and Ng (2012) detected mentions by employing named entity and language-dependent heuristics. They employed multiple sieves (Lee et al., 2011) for English and Chinese, but only used an exact match sieve for Arabic because other sieves did not provide better results. Björkelund and Nugues (2011) considered all noun phrases and possessive pronouns as mentions, and trained two types of classifier: logistic regression and decision trees. Stamborg et al. (2012) extracted all noun phrases, pronouns, and possessive pronouns as mentions. Then they applied (Björkelund and Nugues, 2011)'s solver which consists of various lexical and graph dependency features. Uryupina et al. (2012) adapted for Arabic the BART (Versley et al., 2008) coreference resolution system, which consists of five components: pre-processing pipeline, mention factory, feature extraction module, decoder and encoder. Fernandes et al. (2014) defined a set of rules based on parse tree information to detect mentions, and utilized a latent tree representation to learn coreference chains. Similarly Björkelund and Kuhn (2014) adopted a tree representation approach to cluster mentions, but improved the learning strategy and introduced non-local features to capture more information about coreference relations. There have been other research studies related to anaphora resolution (Trabelsi et al., 2016; Bouzid et al., 2017; Beseiso and Al-Alwani, 2016; Abolohom and Omar, 2015), but they only considered pronominal anaphora. Aloraini and Poesio (2020) also considered a specific type of pronominal anaphora, zero-pronoun anaphora. All current approaches suffer from a number of limitations, one of which is that most of them rely on an extensive set of hand-chosen features.

3 System architecture

3.1 The Baseline System

We use the Lee et al. (2018) system as our baseline and replace their ELMo embeddings with the BERT recipe of Kantor and Globerson (2019a). The input of the system is the concatenated embeddings ($(emb_t)_{t=1}^{T}$) of both word and character levels. The word-level fastText (Bojanowski et al., 2016) and BERT (Devlin et al., 2019) embeddings are used together with the character embeddings learned from a convolution neural network (CNN) during training. The input is then put through a multi-layer bidirectional LSTM to create the token representations ($(x_t)_{t=1}^{T}$). The $(x_t)_{t=1}^{T}$ are used together with head representations (h_i) to form the mention representations (M_i). The h_i of a mention is calculated as the weighted average of its token representations ($\{x_{b_i}, ..., x_{e_i}\}$), where b_i and e_i are the indices of the start and the end of the mention respectively. The mention score ($s_m(i)$) is then computed by a feedforward neural network to determine the likeness of a candidate to be mention. Formally, the system computes h_i, M_i and $s_m(i)$ as follows:

$$\alpha_t = \textsc{ffnn}_\alpha(x_t)$$

$$a_{i,t} = \frac{exp(\alpha_t)}{\sum_{k=b_i}^{e_i} exp(\alpha_k)}$$

$$h_i = \sum_{t=b_i}^{e_i} a_{i,t} \cdot x_t$$
$$M_i = [x_{b_i}, x_{e_i}, h_i, \phi(i)]$$
$$s_m(i) = \text{FFNN}_m(M_i)$$

where $\phi(i)$ is the mention width feature embeddings. To make the task computationally tractable, the system only considers mentions up to a maximum width of 30 tokens (i.e. $e_i - b_i < 30$). Further pruning on candidate mentions is applied before approaching the antecedent selection step. The model keeps a small portion (0.4 mention/token) of the top-ranked spans according to their mention scores ($s_m(i)$).

Next, the system uses a bilinear function to compute a light-weight mention pair scores ($s_c(i,j)$) between all the valid mention pairs[2]. The scores are then used to select top candidate antecedents for all candidate mentions (coarse antecedent selection). More precisely, the $s_c(i,j)$ are computed as follows:

$$s_c(i,j) = M_i^\top W_c M_j$$

After that, the system further computes a more accurate mention pair scores between the mentions and their top candidate antecedents $s_a(i,j)$:

$$P_{(i,j)} = [M_i, M_j, M_i \circ M_j, \phi(i,j)]$$
$$s_a(i,j) = \text{FFNN}_a(P_{(i,j)})$$

where $P_{(i,j)}$ is the mention pair representation, M_i, M_j is the representation of the antecedent and anaphor respectively, $\circ$ denotes element-wise product, and $\phi(i,j)$ is the distance feature between a mention pair.

The next step is to compute the final pairwise score ($s(i,j)$). The system adds an artificial antecedent ϵ to deal with cases of non-mentions, discourse-new mentions or cases when the antecedent does not appear in the candidate list. The $s(i,j)$ is calculated as follows:

$$s(i,j) = \begin{cases} 0 & i = \epsilon \\ s_m(i) + s_m(j) + s_c(i,j) + s_a(i,j) & i \neq \epsilon \end{cases}$$

For each mention the predicted antecedent is the one that has the highest $s(i,j)$. An anaphora-antecedent link will be created only if the predicted antecedent is not ϵ.

Additionally, the model has an option to use higher-order inference to allow the system to access entity level information. We refer the reader to the original Lee et al. (2018) paper for more details. We use the default setting of Lee et al. (2018) to do second-order inference. The final clusters are created using the anaphora-antecedent pairs predicted by the system. Figure 2 shows the proposed system architecture of our system.

3.2 Data Pre-processing

Arabic is a morphologically rich language. Thus, training on Arabic texts that are not pre-processed properly can suffer from sparsity (various forms for the same word) and ambiguity (same form corresponding to multiple words). There are two reasons for these problems. First, certain letters can have different forms which are usually misspelled, such as the various forms of the letter “alif”. Second, the placement of diacritics on words which are assumed to be undiacritized (Habash and Sadat, 2006). Therefore, we follow the steps proposed in (Althobaiti et al., 2014) to pre-process the data. These steps include:

- Normalizing the various forms of the letter ”alif” (إ,أ,آ) to the letter ”ا”.
- Removing all diacritic marks.

We show an example of an original and pre-processed sentence from OntoNotes 5.0 in Table 1. Pre-processing the data increases the overall performance of coreference system with 7 percentage points more as we will see in Section 5.

[2] Candidate mentions are paired with all the mentions appeared before them (candidate antecedents) in the document.

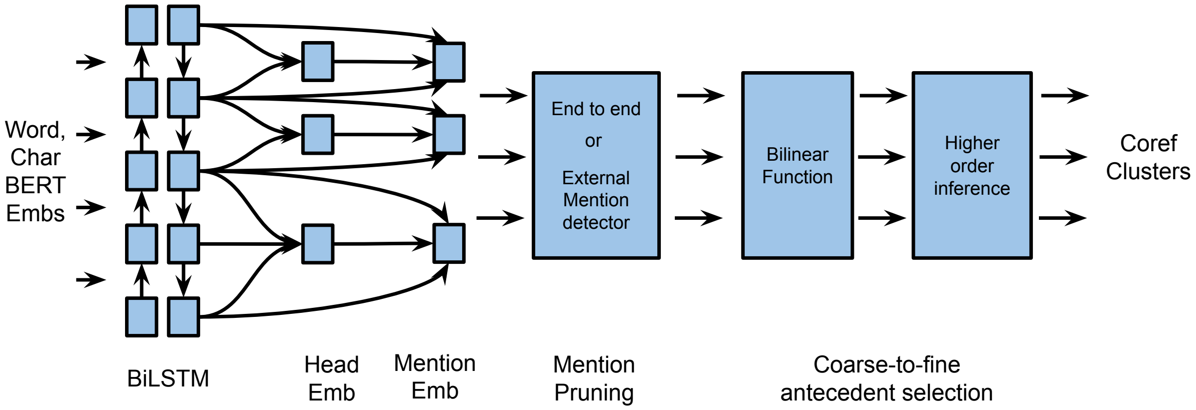

Figure 2: The proposed system architecture.

Original text	إِلى ذلِكَ كَتَبَت مِئاتِ المَقالاتِ النَقْدِيةِ الأَدَبِيةِ
pre-processed text	الى ذلك كتبت مئات المقالات النقدية الادبية

Table 1: An example on how we pre-process Arabic text. The letter ”alif” is normalized and all diacritic marks are removed.

3.3 Multilingual vs. monolingual BERT

BERT (Devlin et al., 2019) is a language representation model consisting of multiple stacked Transformers (Vaswani et al., 2017). BERT was pretrained on a large amount of unlabeled text, and produces distributional vectors for words and contexts. Recently, it has been shown that BERT can capture structural properties of a language, such as its surface, semantic, and syntactic aspects (Jawahar et al., 2019) which seems related to what we need for the coreference resolution. Therefore, we set BERT to produce embeddings for the mentions. BERT is available for English, Chinese, and there is a version for multiple languages, called multilingual BERT [3]. Multilingual BERT is publicly available and covers a wide range of languages including Arabic. Even though the multilingual version provides great results for many languages, it has been shown their monolingual counterparts to achieve better. Therefore, recent research adopts the monolingual approach to pretrain BERT, developing, e.g., CamemBERT for French (Martin et al., 2019), AlBERTo for Italian (Polignano et al., 2019), and others (Lee et al., 2020; Souza et al., 2019; Kuratov and Arkhipov, 2019). AraBERT (Antoun et al., 2020) is a monolingual BERT model for Arabic which was pre-trained on a collection of Wikipedia and newspaper articles. There are two versions, AraBERT 0.1 and AraBERT 1.0, the difference being that the latter pretrained on the word morphemes obtained using Farasa (Darwish and Mubarak, 2016). The two versions yield relatively similar scores in various NLP tasks. In our experiments, we used AraBERT 0.1 because empirically it proved more compatible with the coreference resolution system.

3.4 Mention Detection

Mention detection is a crucial part of the coreference resolution system, better candidate mentions usually lead to better overall performance. As suggested by Yu et al. (2020a), a separately trained mention detector can achieve a better mention detection performance when compared to its end-to-end counterpart. In this work, we adapt the state-of-the-art mention detector of Yu et al. (2020a) to aid our system. In their paper, Yu et al. (2020a) evaluated three different architectures for English mention detection task, we use their best settings (BIAFFINE MD) and replace their ELMo embeddings with BERT embeddings in the same way

[3] https://github.com/google-research/bert

Algorithm 1: End-to-end annealing algorithm.

```
Input: Training step: $N$; Candidate mentions from external mention detector: CANDIDATE_EXTERNAL
Output: Trainable variables: $W$
$n = 0$;
while $n \leq N$ do
    PIPELINE_RATIO $\leftarrow n/N$;
    $rand = random.random()$;
    if $rand \leq$ PIPELINE_RATIO then
        CANDIDATEMENTION $\leftarrow$ CANDIDATE_EXTERNAL;
    else
        Generate mention candidates CANDIDATE_END-TO-END;
        CANDIDATEMENTION $\leftarrow$ CANDIDATE_END-TO-END;
    end
    Predict antecedent for candidate mentions;
    Compute training loss;
    Update $W$;
    $n \leftarrow n + 1$
end
```

Models	Joint			Separate		
	R	P	F1	R	P	F1
BASELINE (MULTIBERT)	85.6	24.4	38.0	**88.1**	25.2	39.2
MULTIBERT+PRE	91.2	26.0	40.5	**93.3**	26.6	41.5
ARABERT+PRE	92.5	26.4	41.1	**95.5**	27.2	42.4

Table 2: The mention detection performance comparison between the separately and jointly trained mention detectors in a high recall setting.

we did for our coreference system[4]. The BIAFFINE MD uses contextual word embeddings and a multi-layer bi-directional LSTM to encode the tokens. It then uses a biaffine classifier (Dozat and Manning, 2017) to assign every possible span in the sentence a score. Finally, the candidate mentions are chosen according to their scores. In addition to the standard high-F1 setting, the system has a further option (high-recall) to output top mentions in the proportion of the number of tokens, this is similar to our mention detection part of the system. Here we use the high-recall settings of the mention detector we modify the baseline system to allow the system using the mentions supplied by the external mention detector.

To confirm our hypothesis that a separately trained mention detector can achieve a better mention detection performance, we compare the mention detection performance of our system with the separately trained mention detector. For our system, we train the models end-to-end and assess the quality of candidate mentions before feeding them into the mention clustering part of the system. Table 2 shows the comparison of both systems in three different settings (MULTIBERT (baseline), MULTIBERT+PRE (multilingual BERT and data pre-processing), ARABERT+PRE (ARABERT and data pre-processing)). As we can see from the table, the separately trained mention detector constantly have a better recall of up to 3% when compared with the jointly trained mention detector[5].

The preliminary experiments show that by simply using the mentions generated by the external mention detector in a pipeline setting result in a lower coreference resolution performance. We believe this is mainly because in an end-to-end setting, the model is exposed to different negative mention examples;

[4] We tried to add the fastText and character-based embeddings to the system but found they do not improve the mention detection results

[5] Here we only care about the recall as the number of candidate mentions is fixed

Category	Training	Dev	Test
Documents	359	44	44
Sentences	7,422	950	1,003
Words	264,589	30,942	30,935

Table 3: Statistics on Arabic portion of CONLL-2012.

Parameter	**Value**
bi-directional LSTM layers/size/dropout	3/200/0.4
FFNN layers/size/dropout	2/150/0.2
CNN filter widths/size	[3,4,5]/50
Char/fastText/Feature embedding size	8/300/20
BERT embedding size/layer	768/Last 4
Embedding dropout	0.5
Max span width	30
Max num of antecedents	50
Mention/token ratio	0.4
Optimiser	Adam (1e-3)
Training step	400K

Table 4: Hyperparameters for our models.

hence, has a better ability to handle false positive candidates. To leverage the benefits between better candidate mentions and more negative mention examples, we introduce a new hybrid training strategy (end-to-end annealing) that initially training the system in an end-to-end fashion and linearly decreasing the usage of end-to-end approach. At the end of the training, the system is trained purely in a pipeline fashion. The resulted system is then tested in a pipeline fashion. Algorithm 1 shows the details of our end-to-end annealing training strategy.

4 Experimental Setup

Since the BERT models are large, the fine-tuning approaches are more computationally expensive: GPU/TPUs with large memory (32GB+) are required. In this work, we use BERT embeddings in a pre-trained fashion to make our experiment feasible on a GTX-1080Ti GPU with 11GB memory.

4.1 Dataset

We run our model on the Arabic portion of OntoNotes 5.0, which were used in the the official CONLL-2012 shared task (Pradhan et al., 2012). The data is divided into three splits: train, development, and test. We used each split for its purpose, the train for training the model, the development for optimizing the settings, and the test for evaluating the overall performance. Detailed information about the number of documents, sentences, and words can be found in Table 3.

4.2 Evaluation Metrics

For our evaluation on the coreference system, we use the official CONLL 2012 scoring script v8.01 to score our predictions. 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. For our experiments on the mention detection we report recall, precision and F1 scores for mentions.

4.3 Hyperparameters

We use the default settings of Lee et al. (2018), and replace their GloVe/ELMo embeddings with the fastText/BERT embeddings. Table 4 shows the hyperparameters used in our system.

Models	MUC			B^3			$CEAF_{\phi_4}$			Avg.
	R	P	F1	R	P	F1	R	P	F1	F1
Björkelund and Nugues (2011)	43.9	52.5	47.8	35.7	49.8	41.6	40.5	41.9	41.2	43.5
Fernandes et al. (2012)	43.6	49.7	46.5	38.4	47.7	42.5	48.2	45.0	46.5	45.2
Björkelund and Kuhn (2014)	47.5	53.3	50.3	44.1	49.3	46.6	49.2	49.5	49.3	48.7
BASELINE (MULTIBERT)	45.7	66.9	54.3	38.8	64.3	48.4	45.7	57.9	51.1	51.3
MULTIBERT+PRE	56.1	67.1	61.1	50.0	63.4	56.0	54.8	61.1	57.8	58.3
ARABERT+PRE	62.3	70.8	66.3	56.3	65.8	60.7	58.8	**66.1**	62.2	63.1
ARABERT+PRE+MD	**63.2**	**70.9**	**66.8**	**57.1**	**66.3**	**61.3**	**61.6**	65.5	**63.5**	**63.9**

Table 5: Coreference resolution results on Arabic test set.

Models	R	P	F1
BASELINE (MULTIBERT)	56.5	79.1	65.9
MULTIBERT+PRE	67.4	78.8	72.6
ARABERT+PRE	70.6	79.9	75.0
ARABERT+PRE+MD	**72.9**	**80.4**	**76.4**

Table 6: Mention detection results on Arabic test set.

5 Evaluation

5.1 Results

Baseline We first evaluate our baseline system using the un-pre-processed data and the multilingual BERT model. As we can see from Table 5, the baseline system already outperforms the previous state-of-the-art system which is based on handcrafted features by a large margin of 2.6 percentage points. The better F1 scores are mainly as a result of a much better precision in all three metrics evaluated, the recall is lower than the previous state-of-the-art system (Björkelund and Kuhn, 2014).

Data pre-processing We then apply heuristic rules to pre-process the data. The goal of pre-processing is to reduce the sparsity of the data by normalizing the letters that have different forms and removing the diacritics. By doing so, we created a 'clean' version of the data. As we can see from Table 5, the simple pre-processing on the data achieved a large gain of 7 percentage points when compared with the baseline model trained on the original data. Since the pre-processing largely reduced the data sparsity, the recall of all three matrices has been largely improved. We further compare the mention scores of two models (see Table 6). As illustrated in the table, the system trained on the pre-processed data achieved a much better recall and a similar precision when compared with the baseline. This suggests that data pre-processing is an efficient and effective way to improve the performance of the Arabic coreference resolution task.

Language Specific BERT Embeddings Next, we evaluate the effect of the language-specific BERT embeddings. The monolingual BERT model (ARABERT) trained specifically on Arabic Wikipedia and several news corpora has been shown that it can outperform the multilingual BERT model on several NLP tasks for Arabic. Here we replace the multilingual BERT model with the ARABERT model to generate the pre-trained word embeddings. We test our system with ARABERT on the pre-processed text, the results are shown in Table 5 and Table 6. As we can see from the Tables, the model enhanced by the ARABERT achieved large gains of 4.7 and 2.4 percentage points when compared to the model using multilingual BERT on coreference resolution and mention detection respectively. Both recall and precision are improved for all the metrics evaluated which confirmed the finding in Antoun et al. (2020) that ARABERT model is better suited for Arabic NLP tasks.

External Mention Detector Finally, we use a separately trained mention detector to guide our models with a better candidate mentions. We train a mention detector using the same CONLL 2012 Arabic datasets and store the top-ranked mentions in the file. We use the top-ranked mentions from the external mention

Corpora	Language	Tokens	Documents
ACE	English	~960,000	-
	Chinese	~615,000	-
	Arabic	~500,000	-
OntoNotes	English	~1,600,000	2384
	Chinese	~950,000	1729
	Arabic	~300,000	447

Table 7: General domain coreference resolution corpora that include Arabic.

detector in a pipeline fashion, the mentions are fixed during the training of the coreference resolution task. We use the output of the mention detector model trained on the pre-processed data and using the AraBERT embeddings as this model performs best over three settings we tested (see Table 2). We use the end-to-end annealing training strategy proposed in Section 3.4 to train our model with both end-to-end and pipeline approaches. The model is then tested in a pipeline fashion. Table 5 shows our results on coreference resolution, the model enhanced by the external mention detector achieved a gain of 0.8% when compared to the pure end-to-end model. We further compared the mention detection performance between two models in Table 6, as expected the new model has a much better mention recall (2.3%) when compared to the pure end-to-end model (araBert+pre), this suggests our training strategy successfully transferred the higher recall achieved by the external mention detector to our coreference system.

Overall, our best model enhanced by the data pre-processing, monolingual Arabic bert and the external mention detector achieved a conll F1 score of 63.9% and this is 15.2 percentage points better than the previous state-of-the-art system (Björkelund and Kuhn, 2014) on Arabic coreference resolution.

5.2 Discussion

Coreference resolution is a difficult task, and even more so for languages such as Arabic with more limited resourced. The main challenge is the lack of large scale coreference resolution corpora. At present there are two multilingual coreference corpora that cover Arabic. The first is the Automatic Content Extraction (ACE) (Doddington et al., 2004) which has ~500,000 tokens, but mentions are restricted to seven semantic types[6] and some can be singletons (mentions that do not corefer). The second is OntoNotes (Pradhan et al., 2012), which covers all entities and does not consider singletons, but the size is smaller than ACE, with ~300,000 tokens. A summary of the two corpora in Table 7. OntoNotes has been the standard for coreference resolution evaluation since the conll-2012 shared task. However, its Arabic portion is small and this scarcity poses a considerable barrier to improving coreference resolution.

Another challenge of the task is the absence of large pre-trained language models. There are two versions of bert: bert-base and bert-large. bert-large integrates more parameters to encode better representations for mentions which usually leads to a better performance in many nlp tasks. AraBERT and multilingual bert are pre-trained using the bert-base approach because bert-large is computationally expensive. We are not aware of any publicly available bert-large for Arabic that we could have used in our experiments. We surmise that a bert-large version of Arabic can improve the overall performance as shown in prior works (Joshi et al., 2019b; Kantor and Globerson, 2019b).

6 Conclusion

In this paper, we modernize the Arabic coreference resolution task by adapting state-of-the-art English coreference system to the Arabic language. We start with a strong baseline system and introduce three methods (data pre-processing, language-specific bert, external mention detector) to effectively enhance the performance of the Arabic coreference resolution. Our final system enhanced by all three methods achieved a conll F1 score of 63.9% and improved the state-of-the-art result on Arabic coreference resolution task by more than 15 percentage points.

[6] The semantic types are person, organization, geo-political entity, location, facility, vehicle, and weapon.

Acknowledgements

This research was supported in part by the DALI project, ERC Grant 695662, in part by the Human Rights in the Era of Big Data and Technology (HRBDT) project, ESRC grant ES/M010236/1.

References

Abdullatif Abolohom and Nazlia Omar. 2015. A hybrid approach to pronominal anaphora resolution in arabic. *Journal of Computer Science*, 11(5):764.

Abdulrahman Aloraini and Massimo Poesio. 2020. Cross-lingual zero pronoun resolution. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 90–98.

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2014. Aranlp: A java-based library for the processing of arabic text.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Majdi Beseiso and Abdulkareem Al-Alwani. 2016. A coreference resolution approach using morphological features in arabic. *International Journal of Advanced Computer Science and Applications*, 7(10):107–113.

Anders Björkelund and Pierre Nugues. 2011. Exploring lexicalized features for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 45–50.

Anders Björkelund and Jonas Kuhn. 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)*, pages 47–57.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Saoussen Mathlouthi Bouzid, Fériel Ben Fraj Trabelsi, and Chiraz Ben Othmane Zribi. 2017. How to combine salience factors for arabic pronoun anaphora resolution. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 929–936. IEEE.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 56–63.

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

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

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

Walter Daelemans, Jakub Zavrel, Kurt Van Der Sloot, and Antal Van den Bosch. 2004. Timbl: Tilburg memory-based learner. *Tilburg University*.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074.

H. Daume and D. Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proc. HLT/EMNLP*, Vancouver.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Timothy Dozat and Christopher Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 41–48, Jeju Island, Korea, July. Association for Computational Linguistics.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Milidiú. 2014. Latent trees for coreference resolution. In *Computational Linguistics, 40(4)*, pages 801–835.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019a. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 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.

Ben Kantor and Amir Globerson. 2019a. 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.

Ben Kantor and Amir Globerson. 2019b. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *CONLL Shared Task '11 Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Sangah Lee, Hansol Jang, Yunmee Baik, Suzi Park, and Hyopil Shin. 2020. Kr-bert: A small-scale korean-specific language model. *arXiv preprint arXiv:2008.03979*.

Baoli li. 2012. Learning to model multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL2012-Shared Task*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

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

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CLiC-it*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task. Association for Computational Linguistics, Association for Computational Linguistics.*, pages 1–40.

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

Fabio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Marcus Stamborg, Dennis Medved, Peter Exner, and Pierre Nugues. 2012. Using syntactic dependencies to solve coreferences. In *Joint Conference on EMNLP and CoNLL2012-Shared Task*.

Fériel Ben Fraj Trabelsi, Chiraz Ben Othmane Zribi, and Saoussen Mathlouthi. 2016. Arabic anaphora resolution using markov decision process. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 520–532. Springer.

Olga Uryupina, Alessandro Moschitti, and Massimo Poesio. 2012. Bart goes multilingual: the unitn/essex submission to the conll-2012 shared task. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 122–128.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. Bart: A modular toolkit for coreference resolution. In *Proceedings of the ACL-08: HLT Demo Session*, pages 9–12.

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

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

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online, July. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020a. Neural mention detection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1–10, Marseille, France, May. European Language Resources Association.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020b. A cluster ranking model for full anaphora resolution. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 11–20, Marseille, France, May. European Language Resources Association.

Desislava Zhekova and Sandra Kübler. 2010. Ubiu: A language-independent system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, page 96–99.