Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 90–98
Marseille, 11–16 May 2020

Cross-lingual Zero Pronoun Resolution

Abdulrahman Aloraini[1,2], **Massimo Poesio**[1]
[1]School of Electronic Engineering and Computer Science, Queen Mary University of London
[2]Department of Information Technology, Qassim University
{a.aloraini, m.poesio}@qmul.ac.uk

Abstract

In languages like Arabic, Chinese, Italian, Japanese, Korean, Portuguese, Spanish, and many others, predicate arguments in certain syntactic positions are not realized instead of being realized as overt pronouns, and are thus called zero- or null-pronouns. Identifying and resolving such omitted arguments is crucial to machine translation, information extraction and other NLP tasks, but depends heavily on semantic coherence and lexical relationships. We propose a BERT-based cross-lingual model for zero pronoun resolution, and evaluate it on the Arabic and Chinese portions of OntoNotes 5.0. As far as we know, ours is the first neural model of zero-pronoun resolution for Arabic; and our model also outperforms the state-of-the-art for Chinese. In the paper we also evaluate BERT feature extraction and fine-tune models on the task, and compare them with our model. We also report on an investigation of BERT layers indicating which layer encodes the most suitable representation for the task. Our code is available at https://github.com/amaloraini/cross-lingual-ZP.

Keywords: zero pronouns, cross-lingual classification, anaphora resolution

1. Introduction

In pronoun-dropping (pro-drop) languages such as Arabic (Eid, 1983), Chinese (Li and Thompson, 1979), Italian (Di Eugenio, 1990) and other romance languages (e.g., Portuguese, Spanish), Japanese (Kameyama, 1985), and others (Kim, 2000), arguments can be elided in certain contexts in which a pronoun is used in English, such as subjects. We use the term zero-pronouns (ZP) to refer to these unrealised arguments, the most used in the recent literature.[1]
Anaphoric zero-pronoun (AZP) are zero-pronouns that refer to one or more noun phrases that appear previously in a text. The following example of an AZP comes from the Arabic section of OntoNotes:

بيان الحريري تميز بتفاصيل . . . حيث ركز * على أن مجلس
وزراء لبنان وحده المسؤول عن . . .

Alhariri's statement included more details ...in which (he) emphasized that the council of ministers of Lebanon is the only representative ...

In the example, the zero pronoun indicated with '*' refers to an entity introduced with a masculine singular noun that was previously mentioned in the sentence. (In OntoNotes 5.0, zero pronouns are denoted as * in Arabic text, and ***pro*** in Chinese).
AZP resolution usually consists of two steps: extracting ZPs that are anaphoric, and identifying the correct antecedents for AZPs. Our focus is on the latter because there has been no proposal for Arabic. In this paper we propose a cross-lingual, BERT based model of zero pronoun resolution. Our contributions include:

- We propose a novel cross-lingual, BERT-based model and test it on languages that differ completely in their morphological structure: Arabic and Chinese. (Arabic is morphologically rich, whereas Chinese's morphology is relatively simple (Pradhan et al., 2012))
- As far as we know this is the first neural network-based ZP resolution model for Arabic, and outperforms the current state-of-the-art on Chinese.
- We carried out an extensive analysis on BERT layers, and discuss which settings can give the optimal performance.

The rest of the paper is organized as follows. We discuss Arabic and Chinese ZP-related literature and in other languages in Section 2. We explain our proposed model in Section 3. We discuss the evaluation settings and results in Section 4. We conclude in Section 5.

2. Related work

2.1. Zero Pronoun Resolution

AZP resolution is included in some coreference resolution systems (Taira et al., 2008; Imamura et al., 2009; Watanabe et al., 2010; Poesio et al., 2010; Yoshino et al., 2013). However, it has proven challenging to combine the task with the resolution of overt mentions, so separating the task from coreference resolution may lead to more improvements (Iida and Poesio, 2011).
Chinese: The release of OntoNotes has spurred a lot of research on zero pronoun resolution in Chinese, but earlier research exists as well. Converse (2006) proposed a rule-based approach that employed Hobbs algorithm (Hobbs, 1978) to resolve ZPs in the Chinese Treebank. Yeh and Chen (2006) is another rule-based approach, using rules from Centering Theory (Grosz et al., 1995). Zhao and Ng (2007), the first machine learning approach to Chinese ZPs, used decision trees and a set of syntactic and positional features. Chen and Ng (2013) extended (Zhao and Ng, 2007) by incorporating contextual features and ZP links. Chen and Ng (2014; Chen and Ng (2015) proposed unsupervised techniques to resolve the task. Kong and Zhou (2010) proposed a tree kernel-based unified framework for ZP detection and resolution. Recent approaches applying deep-learning neural networks include Chen and Ng (2016), the first to apply a forward neural network to the task; Yin et al. (2016), who employed an LSTM to represent AZP and two subnetworks (general encoder and local encoder) to capture

[1] The terms null-subject or zero-subject are also used.

context-level and word-level information of the candidates; Yin et al. (2017), who proposed a deep memory network capable to improve the semantic information of ZPs and its candidates; and Liu et al. (2017), using an attention-based neural network and enhanced the performance by training the model on automatically generated large-scale training data of resolved ZP. Yin et al. (2018), the current state of the art, also used an attention-based model, but combined their network with (Chen and Ng, 2016) features.

Other languages: There has been also a great deal of research on ZPs particularly in Japanese (Kim and Ehara., 1995; Aone and Bennett, 1995; Seki et al., 2002; Isozaki and Hirao, 2003; Iida et al., 2006; Iida et al., 2007; Sasano et al., 2008; Sasano et al., 2009; Sasano and Kurohashi, 2011; Yoshikawa et al., 2011; Hangyo et al., 2013; Iida et al., 2015; Yoshino et al., 2013; Yamashiro et al., 2018), but also in other languages, including Korean (Han, 2004; Byron et al., 2006), Spanish (Ferrández and Peral, 2000), Romanian (Mihăilă et al., 2011), Bulgarian (Grigorova, 2013), and Sanskrit (Gopal and Jha, 2017). Iida and Poesio (2011) proposed the first cross-lingual approach for this task. They used the ILP model of Denis and Baldridge (2007) and introduced a new set of constraints incorporating common features for Italian and Japanese.

All current approaches suffer from a number of limitations, one of which is that most of them rely on an extensive set of features which, as we will see below, are language-dependent. The systems using more complex linguistic features also require larger training datasets than available for many languages, including, e.g., Arabic.

2.2. Arabic

There have been several studies of Arabic coreference resolution task, but none specifically devoted to ZPs except as part of the overall coreference task. In particular, several of the systems involved in the CONLL 2012 shared task attempted Arabic as well. Fernandes et al. (2014) utilized latent tree to capture hidden structure and finding coreference chains. Björkelund and Kuhn (2014) stacked multiple pairwise coreference resolvers and combined decoders to cluster mentions together. Chen and Ng (2012) employed multiple sieves (Lee et al., 2011) for English and Chinese, but used only an exact match sieve for Arabic. Green et al. (2009) proposed CRF sequence classifier to detect Arabic noun phrases, and captured ZPs implicitly. Gabbard (2010) showed that Arabic ZPs can be identified and retrieved. As far as we know none of these proposals reported the results of ZP resolution.

3. Our Model

ZPs resolution involves complex, comprehensive language understanding skills. Resolving ZPs in Chinese requires reasoning, context, and background knowledge of real world entities (Huang, 1984), whereas Arabic, in addition to the previously mentioned skills, requires deep understanding of its rich morphology (Alnajadat, 2017). Recently, it has been shown that BERT (Devlin et al., 2018) 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 resolving ZPs. Therefore, we use BERT to produce a mention representation for AZPs and the candidates, and we also incorporate a few, non language-dependent, features.

Our model is a pairwise classifier classifying <AZP,candidate> pairs to true or false for each of a ZP's candidate antecedents. In this section, we first give an overview of the BERT architecture and its adaption modes. We then describe how we represent the mentions, and how we generate AZP candidates. Finally, we present the hyperparameter tuning and training objective.

3.1. BERT

BERT is a language representation model consisting of stacked multiple Transformers (Vaswani et al., 2017), which can be pretrained on a large amount of unlabeled text, and produces distributional vectors (also called embeddings) for words and contexts. There are several versions of BERT; we use BERT-base Multilingual which was pretrained on many languages, including Chinese and Arabic, and is publicly available[2]. BERT-base Multilingual consists of 12 hidden layers. Each has 768 hidden units and multiple attention heads. Thus, for every input, BERT computes 12 embeddings each of size 768 units.

BERT requires a special format for its input; therefore, it comes with a tool to preprocess its input called Tokenizer. The core of Tokenizer is Wordpiece (Wu et al., 2016) which segments words into sub-words (sub-tokens). Tokenizer also tags the inputs with [CLS] at the beginning and [SEP] at the end. [CLS] is a context classification token made by aggregating the word embeddings in the sentence, and [SEP] indicates the end of a sentence input. An illustration of Tokenizer is shown in Figure 1. The input "*My sweetheart is sleeping*" is preprocessed through Tokenizer. Character sequences *My* and *is* each translates into one token, whereas *sweetheart* and *sleeping* originate two sub-tokens. After the Tokenizer step, tokens are evaluated in BERT which produces their embeddings each of 768 hidden units.

BERT has two modes of adaptation: feature extraction and fine-tuning. Feature extraction (also called feature-based) is when BERT's weights are fixed and used to produce the pretrained embeddings. Fine-tuning is the process of slightly adjusting BERT's parameters for a target task. Both have benefits. Feature extraction is computationally cheaper and might be more suitable for a specific task. Fine-tuning is more convenient to utilize, and may smoothly adapt to several general-purpose tasks. Both modes learn interesting properties about a language and work well for various NLP problems. However, they might not be able to achieve optimal performance for some tasks.

In this paper, we propose combining BERT representations with additional task-related features to improve ZP resolution. In our model, we use BERT feature extraction mode to produce embeddings for AZPs and their antecedents, and add two features: *same_sentence* and *find_distance*. *same_sentence* feature finds whether an AZP and a candidate appear in the same sentence or not, and *find_distance*

[2] https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip

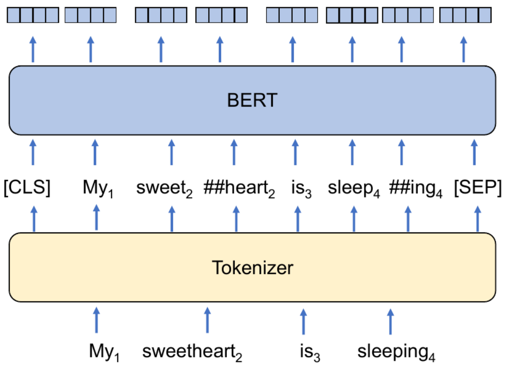

Figure 1: The sentence ”My sweetheart is sleeping” preprocessed through Tokenizer. Tokenizer segments words, and introduces ’[CLS]’ and ’[SEP]’ tokens. After the Tokenizer step, the input is fed into BERT, which outputs embeddings.

computes the word distance between an AZP and a candidate. These two features are cross-lingual and highly related to the task because AZPs and their antecedents usually appear near each other (Chen and Ng, 2014).

3.2. Input representation

Consider a sentence consisting of *n* words and containing an AZP mention at position *i*, so that its previous word is at position *i-1*, and the next word at *i+1*. Let us assume we also have a candidate$_k$ starting at position *k*, and appearing before the AZP.[3] There can be a number of candidates, each of which is a noun phrase.

$$sentence = (w_1, w_2, ..., w_{i-1}, azp_i, w_{i+1}, ..., w_n) \quad (1)$$

$$candidate_k \subset sentence \quad (2)$$

We compute the positional features for every (azp, candidate) pair as follows:
- *same_sentence (azp, candidate)*: returns 1 if an AZP and its candidate are in the same sentence, 0 otherwise.
- *find_distance (azp, candidate)*: finds the word distance between an AZP and its candidate. The word distance is normalized between 0 and 1 based on the training instances.

$$s = same\_sentence(azp_i,\ candidate_k) \quad (3)$$

$$d = find\_distance(azp_i,\ candidate_k) \quad (4)$$

We feed *sentence* into BERT feature extraction mode, which produces the input’s *embeddings*. *embeddings* contain BERT pretrained vectors of every word in *sentence*.

$$embeddings = BERT(sentence) \quad (5)$$

A word can have one representation or several based on the segment step of Tokenizer. For example, in Figure 1 *My* has only one embedding while *sweetheart* has two because it has been segmented into two sub-tokens (sweet and ##heart). In 6, 7, and 8 equations, the subscript of *embeddings* represents the word location in the sentence. μ is a function to compute the mean of a mention representation which can made of several subtoken embeddings [4].

$$a_1 = \mu(embeddings_{(i-1)}) \quad (6)$$

$$a_2 = \mu(embeddings_{(i+1)}) \quad (7)$$

$$c_k = \mu(embeddings_{(k)}) \quad (8)$$

To obtain a mention representation for an AZP, we compute the average embeddings of the AZP previous word and the next word, and join them together. For every candidate, we calculate the mean of its embeddings which then joined with the positional features. We combine the AZP and its candidate representations to form the input to our classifier.

$$azp = [a_1,\ a_2] \quad (9)$$

$$c = [c_k,\ s,\ d] \quad (10)$$

$$input = [azp,\ c] \quad (11)$$

Our classifier consists of multiple multi-layer perceptrons (MLPs) scoring the <azp, candidate> pair ”*input*”.

$$layer_1 = f(W_1 input + b_1) \quad (12)$$

$$layer_2 = f(W_2\, layer_1 + b_2) \quad (13)$$

$$layer_3 = f(W_3\, layer_2 + b_3) \quad (14)$$

$$scoring = f(W_4\, layer_3 + b_4) \quad (15)$$

f is the RELU activation function (Nair and Hinton, 2010). layer$_1$, layer$_2$, layer$_3$, and *scoring* are the resolver’s layers; each has learning parameters *W* and *b*. After scoring all candidates, we choose the candidate with the highest coreference score as the correct antecedent for the AZP. The overall architecture of our model and data representations are shown in Figure 2. In the figure, there is one AZP and two candidates: noun phrase 1 (NP1) and noun phrase 2 (NP2). We run the sentence into BERT to get their word embeddings. AZP is represented with the mean of its previous word, and next word. Candidates are also represented with the mean of their subtoken embeddings, and combined with their positional features. We join each candidate representation with the AZP. We compute <AZP, NP1> and <AZP, NP2> scores which normalized using the softmax layer.

3.3. Candidate generation

For every AZP, we consider as candidate antecedents all maximal and modifier noun phrases (NPs) at most two sentences away, as done by Chen and Ng (2016; Yin et al. (2017). This strategy results in high recall of mentions in both Arabic and Chinese.

3.4. Hyperparameter tuning

We optimize the hyperparameters based on the development sets. We employ three layers and initialize each one’s weights using Glorot and Bengio (2010)’s method. We also add a dropout regularization between every two layers. Table 1 shows the used settings.

[3] An AZP and its candidate may appear in distinct sentences. This could be specified using BERT’s parameters ’text_a’, and ’text_b’. In such cases, however, we empirically found that we get better results by merging the two sentences into one, and add a [SEP] token in between. Thus, we only use ’text_a’.

[4] In our experiments, Tokenizer segmented many Arabic text into several sub-tokens, but rarely did segment Chinese.

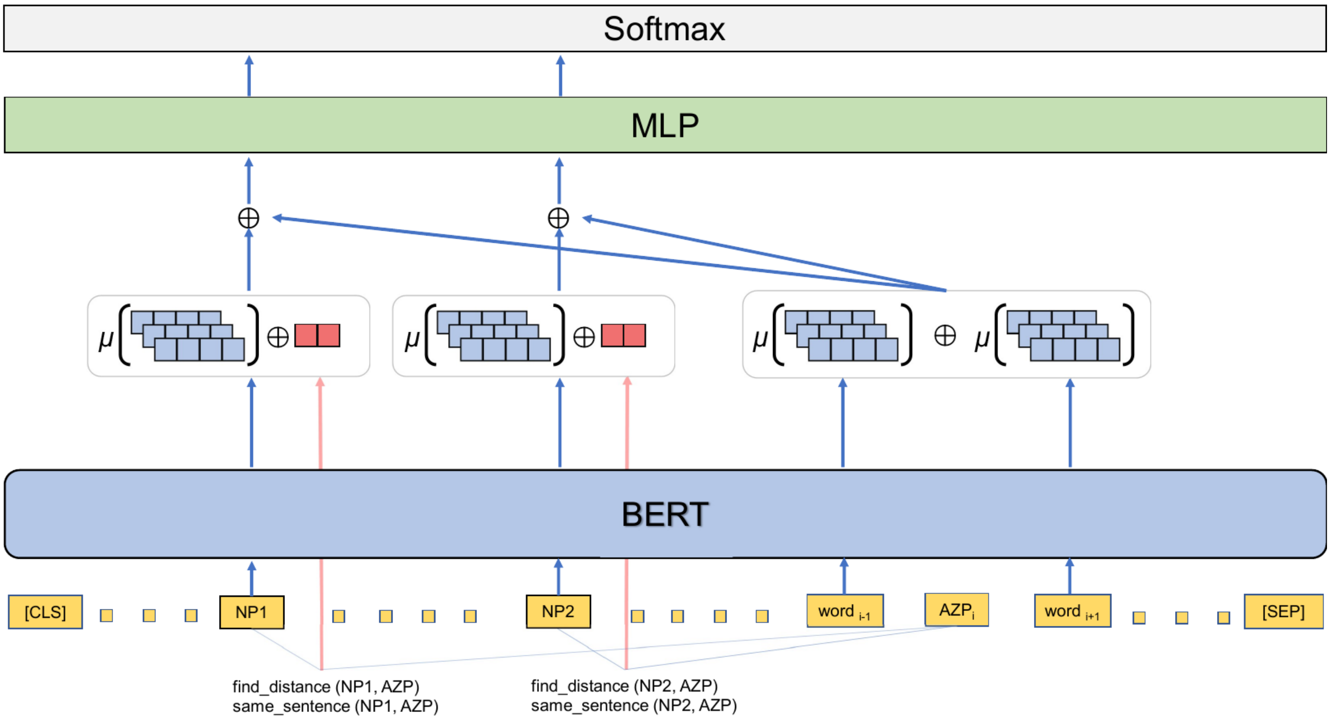

Figure 2: An example of one AZP and two candidates: NP1 and NP2. For every candidate, we calculate its task-specific features *find_distance* and *same_sentence*, the features are represented as 🟥🟥. We compute the average embeddings of each candidate and AZP surrounding words, a subtoken embedding is represented as 🟦🟦🟦🟦. We form <AZP, NP1> and <AZP, NP2> pairs and feed them into a classifier made of MLPs. The classifier finds their scores which then normalized using Softmax. ⊕ is a concatenation operation.

Number of units in the first layer	3300
Number of units in the second layer	2200
Number of units in the third layer	1200
Number of training epochs	10
Learning rate	1e-5
Dropout rate	0.5
Optimizer	Adam

Table 1: Hyperparameter settings.

3.5. Training objective

We minimize the cross entropy error between every AZP and its candidates using:

$$J(\theta) = -\sum_{t \in T}^{n} \sum_{c \in C}^{k} \delta(azp, c) \;\; log(P(azp, c))$$

$$\delta(azp, c) = \begin{cases} 1 & \text{if } c \text{ in } azp \text{ coreference chain} \\ 0 & \text{otherwise} \end{cases}$$

$$\theta = \{W_1, W_2, W_3, W_4, b_1, b_2, b_3, b_4\}$$

θ denotes the set of learning parameters. T consists of the n training instances of AZPs, and C represents the k candidates of an *azp*. *δ(azp, c)* returns whether a candidate *c* is correct antecedent of the *azp*. *log(P(azp, c)* is the predicted log probability of the (*azp*, *c*) pair.

4. Evaluation

4.1. Dataset

We tested our model on the Arabic and Chinese portions of OntoNotes 5.0, which were used in the the official CoNLL-2012 shared task (Pradhan et al., 2012). Gold syntactic parse trees and gold AZPs annotations are available for both languages and were used in all experiments.

Chinese training and development sets contain ZPs, but the test set does not. Therefore, we train the model using the training set and we use the development set as the test set, as done in prior research (Zhao and Ng, 2007; Chen and Ng, 2015; Chen and Ng, 2016; Yin et al., 2016; Yin et al., 2017; Liu et al., 2017; Yin et al., 2018). We hold out 20% of the training data as a development set.

Arabic training, development, and test sets all have ZPs. and we use each set for its purpose. We preprocessed the data by normalizing the letter ”alif” variants and removing all diacritics.

Detailed information about the number of documents, sentences, words, and AZPs can be found in Table 2. The Chinese dataset is larger than Arabic; nonetheless, our model succeeds in resolving many Arabic ZPs.

4.2. Metrics

We evaluate the results in terms of recall, precision, and F-score, defined as in (Zhao and Ng, 2007):

$$Recall = \frac{AZP\ hits}{Number\ of\ AZPs\ in\ Key}$$

Language	Category	Training	Dev	Test
Chinese	Documents	1,391	172	N/A
	Sentences	36,487	6,083	
	Words	756,063	100,034	
	AZPs	12,111	1,713	
Arabic	Documents	359	44	44
	Sentences	7,422	950	1,003
	Words	264,589	30,942	30,935
	AZPs	3,495	474	412

Table 2: Statistics on Chinese and Arabic datasets. Chinese test portion does not contain zero pronouns; therefore, the development portion is used for evaluation as done in prior works.

$$Precision = \frac{AZP\, hits}{Number\ of\ AZPs\ in\ Response}$$

Key represents the true set of AZP entities in the dataset, and *Response* represents the set of identified AZPs in the model. *AZP hits* mean the total number of AZPs correctly resolved with at least one of its antecedents in the gold coreference chain.

4.3. Results

We compare our results with other published results, and with the results using BERT's two adaptation modes. BERT fine-tuning already has a built-in classification layer on top of the stacked Transformers. The feature extraction mode only produces the learned vectors and needs a framework to be trained on. To do so, we implement a bi-attentive neural network to train feature extraction embeddings and optimize it as done in (Peters et al., 2019) who empirically analyzed fine-tuning and feature extraction modes for a few pretrained models, including BERT. In both modes, we train AZPs and their antecedents without the proposed additional features.

4.3.1. Arabic

We report our results for Arabic in Table 3. Given that there was no existing ZP resolver for Arabic, we implemented (Chen and Ng, 2016)'s model and used it as a baseline in our experiments, as it features an extensive range of syntactic, positional, and grammatical features which were then used in other systems as well (Yin et al., 2018).

However, Table 3 shows that these features did not work well with Arabic. We can think of two likely reasons for this. First, the size of Arabic OntoNotes is small, thus might not have provided enough training data for the learning phase. Second, some of Chen and Ng's features might only apply for Chinese; therefore, they might have hurt the performance rather than helped. Also, (Chen and Ng, 2016)'s model lacked morphological features because Chinese morphology is considered relatively simple. In contrast, Arabic morphology is highly derivational and inflectional, and very important for resolving ZPs. Arabic ZPs are preceded by verbs, and verbs encode information about gender, person, and number. The context of ZPs and their antecedents share similar morphological characteristics.

Model	Recall	Precision	F-score
(Chen and Ng, 2016)	8.1	10.1	8.9
BERT (feature extraction)	47.9	59.5	53.1
BERT (fine-tuning)	50.3	62.5	55.8
Our Model	**51.8**	**64.4**	**57.4**

Table 3: Arabic AZPs results.

Interestingly, BERT seems to be capable of modeling these morphological connections and resolve correctly many AZPs. BERT's feature extraction and fine tuning modes produce F-scores of 53.1% and 55.8%. Our model outperforms BERT both modes and achieves an F-score of 57.4%. The incorporated features seem to help with an increase of 1.6% compared to fine tuning, and 4.3% to feature extraction. These findings suggest that while BERT learns many details of a language, it might also need more information to achieve the optimal performance.

4.3.2. Chinese

Our experimental results for Chinese can be seen in Table 4. The Chinese dataset consists of 6 different categories: Broadcast News (BN), Newswires (NW), Broadcast Conversations (BC), Telephone Conversations (TC), Web Blogs (WB), and Magazines (MZ). The state-of-the-art, attention-based model of Yin et al. (2018) performs better than the others in all categories except TC. The TC category contains many short sentences; perhaps Yin et al's model struggles to learn short size inputs. Our model achieves the best overall F-score of 63.5% outperforming all prior models in all categories except in (NW). Specifically, our approach outperforms the current state-of-the-art F-scores in these categories: 1.9% (MZ), 7.4% (WB), 10% (BN), 3.2% (BC), and 8.9% (TC). Feature extraction and fine-tuning modes report 60.4% and 62.1% respectively. Fine tuning process leads to 1.7% increase than feature extraction. Our model outperforms BERT both modes with an increase of 3.1% and 1.4% compared to feature extraction and fine tuning modes. The results in Chinese (even in Arabic) imply that even though fine tuning can improve ZP resolution; however, defining more task-related features with BERT feature extraction mode can enhance AZP resolution.

Other versions of BERT were pretrained specifically for English and Chinese. Chinese-only BERT performs better than BERT Multilingual on Chinese texts in some NLP tasks, according to BERT authors' Github page[5]. Therefore, it might also improve the results we obtain with Chinese, although of course adopting that model would defeat the purpose of developing a cross-lingual model.

4.3.3. BERT Layers

Numerous studies show that BERT layers encode rich information about language structure (Jawahar et al., 2019; Kovaleva et al., 2019; Aken et al., 2019; Goldberg, 2019; Hewitt and Manning, 2019). For a specific NLP task, some layers may carry more useful information than others. In fact, layers that contain indirect information may not lead

[5] https://github.com/google-research/bert/blob/master/multilingual.md

	NW (84)	MZ (162)	WB (284)	BN (390)	BC (510)	TC (283)	Overall
(Zhao and Ng, 2007)	40.5	28.4	40.1	43.1	44.7	42.8	41.5
(Chen and Ng, 2015)	46.4	39.0	51.8	53.8	49.4	52.7	50.2
(Chen and Ng, 2016)	48.8	41.5	56.3	55.4	50.8	53.1	52.2
(Yin et al., 2016)	50.0	45.0	55.9	53.3	55.3	54.4	53.6
(Yin et al., 2017)	48.8	46.3	59.8	58.4	43.2	54.8	54.9
(Liu et al., 2017)	59.2	51.3	60.5	53.9	55.5	52.9	55.3
(Yin et al., 2018)	**64.3**	52.5	62.0	58.5	57.6	53.2	57.3
BERT (feature extraction)	59.3	48.7	66.0	64.9	57.9	59.5	60.4
BERT (fine-tuning)	61.8	51.8	67.9	66.7	58.7	61.6	62.1
Our model	63.4	**54.4**	**69.4**	**68.5**	**60.8**	**62.1**	**63.5**

Table 4: Our proposed model F scores on Chinese ZPs compared with BERT two modes and other models.

to the optimal performance. Therefore, it is important to investigate the internal layers and find the most transferable representation. We examined every BERT layer's weights for our model, and report their behaviour on Arabic and Chinese in Figure 3. We can see that higher layers produce better F-scores than the lower ones. ZP context and true candidates usually share similar morphological characteristics and semantic relationships and higher layers seem to carry such information.

Therefore, the layers in the last half tend to be more relevant to our task than the layers in the lower half. Generally, F-scores increase as we employ higher layers except when we reach the last two layers. Their slight drops of F-scores might be attributed to BERT training objectives. BERT was trained on masked language modeling (MLM) and next sentence prediction (NSP). Since we are using BERT feature extraction mode, the last layers were optimized on these pretrained tasks. Even though MLM and NSP helped BERT model learn linguistic aspects in the internal and middle layers, it might have made the last layers biased and specific to their objective goals. The third-to-last (10th layer) and fourth-to-last (9th layer) layers achieve almost equal high F-scores in Arabic and Chinese, but we find the third-to-last to provide more stable states. In our model, we set the third-to-last as the base to produce embeddings for AZPs and their candidates.

We also tried combinations of layers to see if they can produce better representations for the task. Table 5 reports the first, last, and third-to-last layer F-scores. We compare their F-scores with two more settings: the weighted sum of the last 4 layers and all of 12 layers. The weighted sum of the 4 layers results in 63.1% for Chinese and 55.2% for Arabic. Chinese F-score decreases only 0.4% and Arabic 2.2% compared to their corresponding third-to-last F-scores. When we calculate the mean of all 12 layers, we get 62.4% and 53.1% for Chinese and Arabic respectively. F-scores drop 1.1% for Chinese and 4.7% for Arabic. The weighted sum of multiple layers did not seem to help improve the ZP resolution task. In both settings, Arabic seems to be more sensitive when several layers involved. Arabic morphology is complex and BERT layers might encode its morpheme interactions in some parts of its layers. Some of these interactions might get lost when multiple layers are weighted sum.

BERT Layer(s)	Chinese	Arabic
Third-to-last layer	**63.5**	**57.4**
Last layer	60.9	55.2
First layer	51.2	40.7
Weighted sum of the last 4 layers	63.1	55.2
Weighted sum of all 12 layers	62.4	53.1

Table 5: F-scores results when we use different BERT layer(s) for token representations.

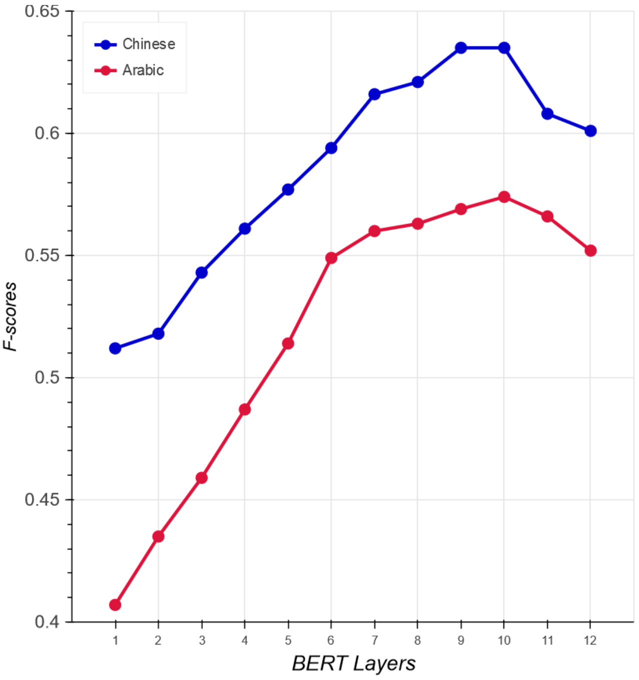

Figure 3: Arabic and Chinese F-scores when we use each of BERT layers to produce mention embeddings. Overall, higher layers produce better representations than the lower layers. The 10th layer led to the highest F-scores in both languages.

5. Conclusion

We presented a cross-lingual model for zero pronoun resolution based on BERT, and evaluated it on the Arabic and Chinese portions of OntoNotes 5.0. Our model is the first to specifically focus on Arabic ZPs, and outperforms state-of-the-art results for Chinese as well. In addition, our model demonstrated better outcomes than BERT fine-tuning and feature extraction modes. We showed that adding positional features to BERT learned representations can improve ZP resolution. We also examined BERT layers, and reported our observations and insights on which layer can be the most suitable for the task. In the future, we plan to develop a ZP identification system, and evaluate our proposed model on more languages and other global features.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions which helped to improve the quality of the paper.

6. Bibliographical References

Aken, B., Winter, B., Löser, A., and Gers, F. A. (2019). How does bert answer questions? a layer-wise analysis of transformer representations. In *arXiv preprint arXiv:1908.08593*.

Alnajadat, B. M. (2017). Pro-drop in standard arabic. In *International Journal of English Linguistics 7.1*.

Aone, C. and Bennett, S. W. (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. In *ACL '95 Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 122–129.

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)*, pages 47–57.

Byron, D. K., Gegg-Harrison, W., and Lee, S.-H. (2006). Resolving zero anaphors and pronouns in korean. In *Traitement Automatique des Langues 46.1*, pages 91–114.

Chen, C. and Ng, V. (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.

Chen, C. and Ng, V. (2013). Chinese zero pronoun resolution: Some recent advances. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1360–1365.

Chen, C. and Ng, V. (2014). Chinese zero pronoun resolution: An unsupervised probabilistic model rivaling supervised resolvers. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Chen, C. and Ng, V. (2015). Chinese zero pronoun resolution: A joint unsupervised discourse-aware model rivaling state-of-the-art resolvers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 320–326.

Chen, C. and Ng, V. (2016). Chinese zero pronoun resolution with deep neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788.

Converse, S. (2006). Pronominal anaphora resolution in chinese. In *PhD Thesis, University of Pennsylvania*.

Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv:1810.04805*.

Di Eugenio, B. (1990). Centering theory and the italian pronominal system. In *Proc. of the 13th COLING*, Helsinki, Finland.

Eid, M. (1983). On the communicative function of subject pronouns in arabic. In *Journal of Linguistics 19.2*, pages 287–303.

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

Ferrández, A. and Peral, J. (2000). A computational approach to zero-pronouns in spanish. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 166–172.

Gabbard, R. (2010). Null element restoration. In *Ph.D Thesis, University of Pennsylvania*.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*.

Goldberg, Y. (2019). Assessing bert's syntactic abilities. In *arXiv preprint arXiv:1901.05287*.

Gopal, M. and Jha, G. N. (2017). Zero pronouns and their resolution in sanskrit texts. In *The International Symposium on Intelligent Systems Technologies and Application*, pages 255–267.

Green, S., Sathi, C., and Manning, C. (2009). Np subject detection in verb-initial arabic clauses. In *Proceedings of the Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3). Vol. 112.*

Grigorova, D. (2013). An algorithm for zero pronoun resolution in bulgarian. In *Proceedings of the 14th International Conference on Computer Systems and Technologies*.

Grosz, B., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. In *Computational linguistics 21, no. 2*, pages 203–225.

Han, N.-R. (2004). A korean null pronouns: Classification and annotation. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation. Association for Computational Linguistics, 2004.*, pages 33–40.

Hangyo, M., Kawahara, D., and Kurohashi, S. (2013). Japanese zero reference resolution considering exophora and author/reader mentions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 924–934.

Hewitt, J. and Manning, C. (2019). A structural probe for finding syntax in word representations. 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)*.

Hobbs, J. (1978). Resolving pronoun references. In *Lingua*, pages 311–338.

Huang, C.-T. J. (1984). On the distribution and reference of empty pronouns. In *Linguistic Inquiry, Vol. 15, No. 4*, pages 531–574.

Iida, R. and Poesio, M. (2011). A cross-lingual ilp solution to zero anaphora resolution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 804–813.

Iida, R., Inui, K., and Matsumoto, Y. (2006). Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistic*, pages 625–632.

Iida, R., Inui, K., and Matsumoto., Y. (2007). Zero-anaphora resolution by learning rich syntactic pattern features. In *ACM Transactions on Asian Language Information Processing, 6(4)*.

Iida, R., Torisawa, K., Hashimoto, C., Oh, J.-H., and Kloetzer, J. (2015). Intra-sentential zero anaphora resolution using subject sharing recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2179–2189.

Imamura, K., Saito, K., and Izumi, T. (2009). Discriminative approach to predicate argument structure analysis with zero-anaphora resolution. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 85–88.

Isozaki, H. and Hirao, T. (2003). Japanese zero pronoun resolution based on ranking rules and machine learning. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing,*, pages 184–191.

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

Kameyama, M. (1985). *Zero Anaphora: The case of Japanese*. Ph.D. thesis, Stanford University, Stanford, CA.

Kim, Y.-B. and Ehara., T. (1995). Zero-subject resolution method based on probabilistic inference with evaluation function. In *Proceedings of the 3rd Natural Language Processing Pacific- Rim Symposium*, pages 721—727.

Kim, Y.-J. (2000). Subject/object drop in the acquisition of korean: A cross-linguistic comparison. In *Journal of East Asian Linguistics 9.4*, pages 325–351.

Kong, F. and Zhou, G. (2010). A tree kernel-based unified framework for chinese zero anaphora resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, pages 882–891.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of bert. In *arXiv preprint arXiv:1908.08593*.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (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.

Li, C. N. and Thompson, S. A. (1979). Third person pronouns and zero anaphora in chinese discourse. In *Syntax and Semantics*, volume 12: Discourse and Syntax, pages 311–335. Academic Press.

Liu, T., Cui, Y., Yin, Q., Zhang, W., Wang, S., and Hu, G. (2017). Generating and exploiting large-scale pseudo training data for zero pronoun resolution. In *arXiv preprint arXiv:1606.01603*.

Mihăilă, C., Ilisei, I., , and Inkpen, D. (2011). Zero pronominal anaphora resolution for the romanian language. In *Research Journal on Computer Science and Computer Engineering with Applications, POLIBITS, 42*.

Nair, V. and Hinton, G. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Peters, M., Ruder, S., and Smith, N. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *arXiv preprint arXiv:1903.05987*.

Poesio, M., Uryupina, O., and Versley, Y. (2010). Creating a coreference resolution system for italian. In *LREC 2010 May 19*.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (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.

Sasano, R. and Kurohashi, S. (2011). discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 758–766.

Sasano, R., Kawahara, D., and Kurohashi, S. (2008). A fully-lexicalized probabilistic model for japanese zero anaphora resolution. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 769–776.

Sasano, R., Kawahara, D., and Kurohashi, S. (2009). The effect of corpus size on case frame acquisition for discourse analysis. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 521–529.

Seki, K., Fujii, A., and Ishikawa., T. (2002). A probabilistic method for analyzing japanese anaphora integrating zero pronoun detection and resolution. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, pages 1–7.

Taira, H., Fujita, S., and Nagata, M. (2008). A japanese predicate argument structure analysis using decision lists. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 523–532.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Watanabe, Y., Asahara, M., and Matsumoto, Y. (2010). A structured model for joint learning of argument roles and predicate senses. In *Proceedings of the ACL 2010 Conference Short Papers,*, pages 98–101.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*.

Yamashiro, S., Nishikawa, H., and Tokunaga, T. (2018). Neural japanese zero anaphora resolution using smoothed large-scale case frames with word embedding. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

Yeh, C.-L. and Chen, Y.-C. (2006). Zero anaphora resolution in chinese with shallow parsing. In *Journal of Chinese Language and Computing 17 (1)*, pages 41–56.

Yin, Q., Zhang, Y., Zhang, W., and Liu, T. (2016). A deep neural network for chinese zero pronoun resolution. In *arXiv preprint arXiv:1604.05800.*

Yin, Q., Zhang, Y., Zhang, W., and Liu, T. (2017). Chinese zero pronoun resolution with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1309–1318.

Yin, Q., Zhang, Y., Zhang, W., Liu, T., and Wang, W. Y. (2018). Zero pronoun resolution with attention-based neural network. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 13–23.

Yoshikawa, K., Asahara, M., and Matsumoto, Y. (2011). Jointly extracting japanese predicate-argument relation with markov logic. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1125–1133.

Yoshino, K., Mori, S., and Kawahara, T. (2013). Predicate argument structure analysis using partially annotated corpora. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 957–961.

Zhao, S. and Ng, H. T. (2007). Identification and resolution of chinese zero pronouns: A machine learning approach. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 541–550.