A Deep Neural Network for Chinese Zero Pronoun Resolution

Yin Qingyu,Zhang Weinan, Zhang Yu, Liu Ting
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
Affiliation / Address line 3
{qyyin, wnzhang, yzhang, tliu}@ir.hit.edu.cn

Abstract

This paper investigates the problem of Chinese zero pronoun resolution. Most existing approaches are based on machine learning algorithms, using hand-crafted features, which is labor-intensive. Moreover, semantic information that is essential in the resolution of noun phrases has not been addressed enough by previous approaches on zero pronoun resolution. This is because that zero pronouns have no descriptive information, which makes it almost impossible to calculate semantic similarity between the zero pronoun and its candidate antecedents. To deal with these problems, we aim at exploring learning algorithms that are capable of generating semantic representations for zero pronouns, capturing the intricate relatedness between zero pronouns and candidate antecedents, and meanwhile less dependent on extensive feature engineering. To achieve this goal, in this paper, we propose a zero pronoun-specific neural network for Chinese zero pronoun resolution task. Experimental results show that our approach significantly outperforms the state-of-the-art method.

1 Introduction

Zero pronoun (ZP) refers to the component that is omitted due to the coherence of language. A ZP can be either anaphoric if it corefers to one or more noun phrases (NPs) in the preceding text, or non-anaphoric if there are no such NPs. These NPs that supply the necessary information for interpreting the gaps are normally described as antecedents. Following shows an example of ZPs and their antecedents, where "ϕ" is used to denote a ZP.

> [当事人 李亚鼎] 除了 表示 ϕ_1 欣然 接受但 ϕ_2 也 希望 国家 要 有 人 负责。
>
> [Litigant Li Yading] not only shows ϕ_1 willing of acception, but also ϕ_2 hopes that there should be someone in charge of it.

The ZP "ϕ_2" in this example is an anaphoric zero pronoun (AZP) that refers to the NP "当事人李亚鼎/Litigant Li Yading" while another ZP "ϕ_1" is non-anaphoric.

ZP is ubiquitous in the pro-drop languages, such as Chinese, Japanese, etc. As mentioned in (Kim, 2000), the use of overt subjects in English is over 96% while the percentage is only 64% in Chinese, indicating that the ZP phenomenon in Chinese is much more prevalent. Meanwhile, important descriptive information such as gender and number that has been proven to be essential in pronoun resolution is absent to ZPs, denotes ZP resolution a non-trivial task. Therefore, it is challenging but also crucial to resolving ZPs, especially for the pro-drop languages such as Chinese.

Recent researches on Chinese ZP resolution can be classified into two types: supervised approaches (Zhao and Ng, 2007; Kong and Zhou, 2010; Chen and Ng, 2013) and unsupervised ones (Chen and Ng, 2014; Chen and Ng, 2015). In these approaches, the task of ZP resolution is regarded as a classification problem. The majority of them follow (Zhao and Ng, 2007), employ machine learning algorithms to build classifiers. In this direction, most studies focus on designing effective features to obtain better classification performance. Even for the unsupervised approach (Chen and Ng, 2014), they utilize a learning-based ranking model by employing 36 features to help resolve anaphoric ZPs. Admittedly, feature engineering is important but labor-intensive. It is therefore desirable to discover explanatory factors

from data and make the learning algorithms less dependent on extensive feature engineering learning algorithms to build classifiers (Bengio, 2013).

In addition, for most of the existing approaches, they utilize only grammatical and syntactic features, ignore semantic information that plays a crucially important role in the resolution of common NPs (Ng, 2007). The main reason is that ZPs have no descriptive information, which makes it almost impossible to calculate semantic similarity and relatedness score between the ZP and its candidate antecedents. Nevertheless, we notice that a ZP's contextual information can help describe it. For example, given a sentence " ϕ is sweet.", one may resolve the ZP "ϕ " to the NP candidate "an apple", but will never regard "a book" as its antecedent, because they naturally look at the ZP's context "is sweet" to resolve it. These motivate us to develop an approach that can effectively model contextual information to generate the semantic representation for a ZP. By these representations, we can evaluate semantic relatedness of the ZP and its NP candidates.

In this paper, we propose a deep learning approach for Chinese ZP resolution. On the basis of the long-short term memory (LSTM), we develop a zero pronoun-specific neural network (**ZPSNN**), which is capable of generating continuous semantic representations for ZPs by utilizing their context words, capturing the intricate semantic relatedness between ZPs and NP candidate, and meanwhile avoiding feature engineering. In this study, an end-to-end ZP resolution system is built to jointly identify and resolve anaphoric ZPs. We conduct our experiments on the OntoNotes 5.0 corpus, comparing with the baseline in different experimental settings. Results show that our approach outperforms the baseline algorithm and achieves state-of-the-art performance.

The major contributions of the work presented in this paper are as follows.

- We develop a zero pronoun-specific neural network for Chinese ZP resolution task, avoiding the feature engineering algorithms.
- We build the **ZPSNN** that is capable of modeling ZP's contextual information to generate the semantic representations for ZPs, by which can we reveal relatedness of a ZP and its NP candidates at the semantic level.
- We empirically verify that our approach outperforms the state-of-the-art approach in Chinese zero pronoun resolution task on the OntoNotes 5.0 corpus.

2 Overview of Our Approach

In this section, we give an overview of our approach for Chinese ZP resolution. For the input texts, we first extract a set of NP candidates for ZPs. Following (Chen and Ng, 2015), to avoid having to deal with a potentially large number of NP candidates, we consider all and only those NPs that are two sentences away at most from the given ZP to be its potential candidate antecedents. In addition, among these NPs, we qualify those who are either maximal NPs or modifier NPs as candidates.

With all these NP candidates selected, ZP resolution then proceeds. In this study, we regard it as a classification process, developing a zero pronoun-specific neural network (**ZPSNN**) to fulfill the classification issue. An illustration of **ZPSNN** is given in Figure 1, details will be presented in the next section. Inputs of the **ZPSNN** consist of two separate parts: one for ZP and another for NP candidate. For a given ZP and NP candidate, **ZPSNN** generates semantic representations for them by employing two recurrent neural networks. **ZPSNN** outputs the classification results of the given ZP and NP candidate into: "positive", which means that the NP candidate is an antecedent of the ZP, or otherwise, "negative". Specifically, as we are building an end-to-end ZP resolution system, which means that AZP identification should be taken into consideration, we add the third classification category "non" that indicates the ZP is predicted to be un-anaphoric.We determine whether a ZP is anaphoric in the following process. For each ZP-NP pair, a classification label is assigned among "positive", "negative" and "non". For a given ZP, we test all its NP candidates, calculate the ratio of the label "non" in all its ZP-NP pairs. A ZP is considered as a non-anaphoric ZP, if its "non" ratio is bigger than 0.5, otherwise an AZP. Once a ZP is defined as an AZP, we then resolve it with an antecedent.

For a given AZP zp, we test each of its NP candidates from right to left. If the pair (zp,np_1) is classified as "positive" by the classifier, we then regard np_1 as the antecedent of zp. Otherwise, we proceed to the next NP np_2 immediately to the left of np_1, and go through the classification procedure again. This process continues until we find

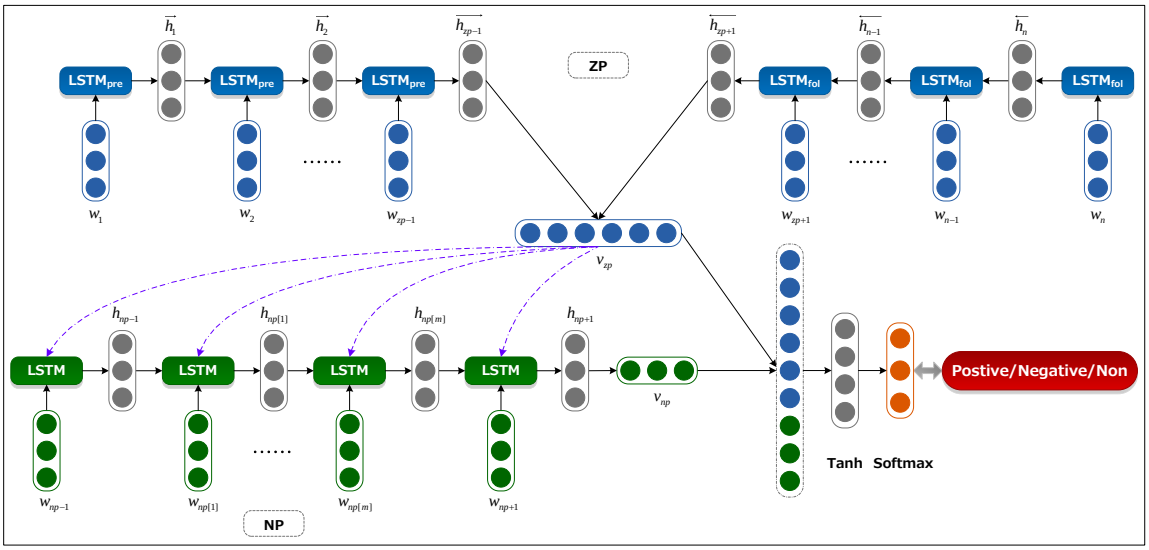

Figure 1: The zero pronoun-specific neural network (**ZPSNN**) for ZP resolution. For ZP part, w_i means the i-th word in the sentence, w_{zp-i} is the i-th last word before the ZP and w_{zp+i} is the i-th word behind the ZP. For NP part, $w_{np[i]}$ denotes the i-th word of the NP, and $w_{np+1(-1)}$ is the word appears immediately after (before) the NP.

an antecedent for zp, or there is no NP candidate to test. Such a closest-first search strategy is also used in (Zhao and Ng, 2007; Chen and Ng, 2013).

3 Zero Pronoun-Specific Neural Network

In this section, we will describe our proposed zero pronoun-specific neural network (**ZPSNN**) for ZP resolution in details.

3.1 Architecture

Figure 1 shows the structure of **ZPSNN**, which consists of two parts, i.e. one for ZP and another for NP candidate. In an attempt to represent ZPs by their contextual information, for each ZP, a sequence of words from the beginning to the end of the sentence it appears are extracted as the input of ZP part. For NP part, its input includes all the words in NP and two additional words appear immediately before and after the NP that are employed as an information expansion for the NP. In order to capture the semantics of words, we represent each word as a low dimensional, continuous and real-valued vector, also known as word embedding (Bengio et al., 2003). All the word vectors are stacked in a word embedding matrix $L_w \in \mathbb{R}^{d \times |V|}$, where d is the dimension of word vector (100 in our experiments) and $|V|$ is the size of word vocabulary.

As the inputs of both parts of **ZPSNN** are sequences of words, one way to encode the input word sequences is via a recurrent neural network (Elman, 1991). Recurrent neural networks (RNN) have been widely exploited to deal with variable-length sequence input. As mentioned in (Mikolov et al., 2010), an RNN can overcome the limitations of traditional model in capturing only a fixed-length history, yielding significant performance improvements in terms of perplexity reduction and speech recognition accuracy. An RNN stores the sentence history in a real-valued history vector, which captures coherence of the sentence. Such a history vector reveals the semantic representation of the sentence (a sequence of words), which denotes the contextual information we expect. LSTM (Hochreiter and Schmidhuber, 1997) is one of the classical variations of RNN to mitigate the gradient vanish problem of RNN. Assume $x = x_1, x_2, ...x_n$ is an input sequence. Each time step t has an input x_t and a hidden state h_t. The internal mechanics of the LSTM is defined as:

$$i_t = \sigma(W^{(i)} \cdot [x_t; h_{t-1}] + b^{(i)}) \quad (1)$$

$$f_t = \sigma(W^{(f)} \cdot [x_t; h_{t-1}] + b^{(f)}) \quad (2)$$

$$o_t = \sigma(W^{(o)} \cdot [x_t; h_{t-1}] + b^{(o)}) \quad (3)$$

$$\tilde{C}_t = tanh(W^{(c)} \cdot [x_t; h_{t-1}] + b^{(c)}) \quad (4)$$

$$C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \quad (5)$$

$$h_t = o_t \odot tanh(C_t) \quad (6)$$

where $\odot$ is an element-wise product, $W^{(i)}$, $b^{(i)}$, $W^{(f)}$, $b^{(f)}$, $W^{(o)}$, $b^{(o)}$, $W^{(c)}$, $b^{(c)}$ are the parameters of the network, and σ is the sigmoid activation function. In the LSTM architecture, three gates are employed to allow the model to suffer less from the vanishing gradient problem. The input gate i determines how incoming vector x influences the memory cell C. The output gate o allows the memory cell to have effects on the outputs. The forget gate f can help the cell to decide to forget or remember its previous state.

ZP-centered LSTM: The aforementioned LSTM suffers a weakness of not utilizing the contextual information from the future tokens. However, as we are going to represent the ZP by its contextual information, both the preceding and following contexts are important. Hence, we make a slight modification to the aforementioned LSTM model, introducing a ZP-centered LSTM for the ZP part in **ZPSNN**. The basic idea is to model both the preceding and following contexts surrounding the ZP, so that contexts in both directions could be utilized to represent the semantic of ZP. Specifically, we employ two LSTM neural networks, LSTM$_{pre}$ and LSTM$_{fol}$, to model the preceding and following contexts respectively. The input of LSTM$_{pre}$ is the preceding contexts of a ZP, and the input of LSTM$_{fol}$ is the following contexts of a ZP. We run LSTM$_{pre}$ from left to right, and run LSTM$_{fol}$ from right to left. Afterwards, we concatenate the last hidden vectors of LSTM$_{pre}$ and LSTM$_{fol}$ together to generate the continuous representation of ZP, and feed it to the next layer to go through the rest procedures of the ZP resolution. One could also try averaging or summing the last hidden vectors of LSTM$_{pre}$ and LSTM$_{fol}$ as alternatives.

Attention Mechanism: In an attempt to dynamically align the more informative parts of NPs to the ZPs, we apply an attention mechanism for our model (the purple dotted lines in Figure 1). This strategy has been widely used in many other nature language processing tasks, such as factoid question answering (Hermann et al., 2015), sentence summarization (Rush et al., 2015) and Entailment (Rocktäschel et al., 2015). In this study, we use a gating function as our attention mechanism. Specifically, given the output of the ZP part $v^{(zp)}$, the output vector and input word vector of LSTM in NP part at time t, $h_t^{(np)}$ and x_t, the attention mechanism computes a gate as: $g_t = G(x_t, h_t^{(np)}, v^{(zp)})$. The scoring function G is defined as:

$$m(a)_t = tanh(W^{(am)}h_t^{(np)} + U^{(am)}x_t + V^{(am)}v^{(zp)} + b^{(am)}) \quad (7)$$

$$g_t = \sigma(W^{(g)}m(a)_t + b^{(g)}) \quad (8)$$

where $W^{(am)}$, $U^{(am)}$, $V^{(am)}$, $W^{(g)}$, $b^{(am)}$ and $b^{(g)}$ are attention parameters. Then, the update vector for each word in NP $\tilde{h}_t^{(np)}$ are formulated as:

$$\tilde{h}_t^{(np)} = h_t^{(np)} g_t \quad (9)$$

Experiments show that the attention models lead to a substantial improvement from non-attentive models, indicating that the attention mechanism can help to select the correct antecedent more efficiently from NP candidates according to the continuous representation of ZP.

After calculating the hidden vectors of both ZP and NP part, we regard the last hidden vectors of each part as the semantic representations of ZP and the NP candidate, respectively. We then concatenate these two vectors, and feed it to a Tanh layer whose output length is class number. Finally, we add a softmax layer to output the probability of classifying the ZP and NP candidate as "postive", "negtive" or "non". Softmax function is calculated as follows, where C is the number of classification categories.

$$softmax_i = \frac{exp(x_i)}{\sum_{i'=1}^{C} exp(x_{i'})} \quad (10)$$

3.2 Training and Initialization

We train the **ZPSNN** in an end-to-end way in a supervised learning framework. Training instances are created as follows. Each training instance corresponds to a ZP and one of its NP candidates. As mentioned, for all the NPs in the parse tree within 2 sentences away at most from the given ZP, we qualify those who are either maximal NPs or modifier NPs as NP candidates. If a ZP is non-anaphoric ZP, we simply give it with its corresponding NP candidate a "non" category, or otherwise, if an NP candidate corefers to the given ZP in the gold standard coreference chain, we regard it as a "positive" training example, if not, a "negative" one.

	Setting 1: Gold Parse + Gold AZP						Setting 2: Gold Parse + System AZP						Setting 3: System Parse + System AZP					
	Baseline			**ZPSNN**$^{\dagger}_{*}$			Baseline			**ZPSNN**$^{\dagger}_{*}$			Baseline			**ZPSNN**$^{\dagger}_{*}$		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Overall	50.0	50.4	50.2	54.7	55.9	**55.3**	35.7	26.2	30.3	46.8	25.5	**33.1**	19.6	15.5	17.3	31.6	24.5	**27.6**
NW	46.4	46.4	**46.4**	45.1	45.1	45.1	32.1	28.1	30.3	33.8	30.9	**32.3**	11.9	14.3	13.0	20.0	19.4	**19.7**
MZ	38.9	39.1	39.0	44.6	44.6	**44.6**	29.6	19.6	23.6	31.2	26.9	**28.9**	4.9	4.7	4.8	12.5	11.9	**12.2**
WB	51.8	51.8	51.8	53.0	53.0	**53.0**	39.1	22.9	28.9	44.6	22.0	**29.5**	20.1	14.3	16.7	34.4	26.3	**29.8**
BN	53.8	53.8	53.8	55.5	55.5	**55.5**	30.8	30.7	**30.7**	48.0	22.1	30.3	18.2	22.3	20.0	24.6	19.0	**21.4**
BC	49.2	49.6	49.4	59.8	60.5	**60.2**	35.9	26.6	30.7	50.8	27.0	**35.3**	19.4	14.6	16.7	35.4	27.6	**31.0**
TC	51.9	53.5	52.7	64.3	66.1	**65.2**	43.5	28.7	34.6	54.9	29.5	**38.4**	31.8	17.0	22.2	50.7	32.0	**39.2**

Table 1: Experimental result on the test data. **ZPSNN** represents our approach. $^{\dagger}_{*}$ indicate that our approach is statistical significant over the baselines (using t-test, with $p < 0.05$).

We regard cross-entropy error between standard classification and predicted classification as the loss function.

$$loss = -\sum_{t \in T} \sum_{c=1}^{C} P_c^g(t) log(P_c(t)) \qquad (11)$$

where T is the training data, t denotes one of training instance, C presents the number of classification categories, $P_c(t)$ is the probability of predicting t as class c given by the **ZPSNN** and $P_c^g(t)$ is a binary-value number, whose value is 1 if c is the correct class of t, otherwise 0. We take the derivative of loss function through back-propagation with respect to the whole set of parameters, and update parameters with stochastic gradient descent. We use *100-dimensional* Chinese word embeddings[1] as inputs. For our parameters, we randomly initialize them from a uniform distribution $U(-0.1, 0.1)$, and set the learning rate as 0.01.

4 Experiments

4.1 Corpus

We employ the dataset used in the official CoNLL-2012 shared task, from the OntoNotes Release 5.0[2], to carry out our experiment. The CoNLL-2012 shared task dataset consists of three parts, i.e. a training set, a development set, and a test set. Table 2 shows the statistics of our corpus. Documents in the corpus come from six sources, namely Broadcast News (BN), Newswires (NW), Broadcast Conversations (BC), Telephone Conversations (TC), Web Blogs (WB), and Magazines (MZ). Considering that only the training set and the development set are annotated with ZPs, we thus utilize the training set for training and the development set for testing. The same experimental data setting is utilized in our baseline system (Chen and Ng, 2015).

	Documents	Sentences	ZPs	AZPs
Train	1,391	36,487	23,065	12,111
Test	172	6,083	3,658	1,713

Table 2: Statistics of our corpus.

4.2 Evaluation metrics

Following researches on zero pronoun resolution (Zhao and Ng, 2007; Chen and Ng, 2015), we evaluate the results in terms of recall (R), precision (P), and F-score (F).

4.3 Experimental results

We employ (Chen and Ng, 2015)'s system as the baseline, which is the state-of-the-art ZP resolution system for Chinese. To evaluate our proposed approach, following (Chen and Ng, 2015), three experimental settings are designed. In Setting 1, we assume that gold AZPs and gold syntactic parse trees are available (obtained from the CoNLL-2012 shared task dataset). In Setting 2, we utilize gold syntactic parse trees and system AZPs, which means that we employ the **ZPSNN** to identify whether a ZP is anaphoric or not. Finally, in Setting 3, we employ system AZP and system syntactic parse trees that obtained by Berkeley parser, which is the state-of-the-art parsing model. Experimental results of our approach and the baseline system are shown in Table 1. The first row in Table 1 is the overall scores, follows are results in different sources in test data. As we can see, our approach significantly outperforms the baseline system under three experimental settings by

[1]Embeddings are trained by **word2vec** toolkit: `https://code.google.com/p/word2vec/` on corpus: `http://www.sogou.com/labs/dl/cs.html`.

[2]`http://catalog.ldc.upenn.edu/LDC2013T19`

5.1%, 2.8%, 10.3% in terms of overall F-score, respectively. In per-source results, for Setting 1 and 2, **ZPSNN** beats the baseline system in five of the six sources of data, only slightly under-performs in one source in each setting. Moreover, in Setting 3, our system obtains extremely higher performance than the baseline system in all the six sources of data. All these approve that our proposed approach achieves a considerable improvement in ZP resolution.

One important thing to note is that **ZPSNN** gains particular great performance in the experimental Setting 3 (an end-to-end experimental setting), where system AZP and system parse trees are employed. This is because that for the baseline system, features are extracted via the syntactic parse trees. Therefore, errors introduced by applying the system parser adversely impact the overall performance. While in **ZPSNN**, syntactic parse trees are only employed to extract NP candidates, thus inaccurate parse tree results can only make a limited bad influence. Consequently, our approach performances much better than the baseline system in the end-to-end experimental setting.

Note that for per-source results, the performance of **ZPSNN** has a great disparity among six sources of corpus. To gain additional insight into the differences in performance, we make a statistic on each source of corpus, as shown in Table 3.

	CandiNum	AntDis	AveLen	AZP/ZP
NW	18.55	3.42	24.8	77.6%
MZ	23.8	4.5	34.4	80.1%
WB	12.4	1.89	19.2	47.0%
BN	22.3	3.26	32.1	48.6%
BC	7.8	0.99	16.0	44.9%
TC	4.1	0.44	10.1	44.0%

Table 3: Statistics of six sources of corpus. **CandiNum** is the average candidate number of each AZP; **AntDis** presents the average distance (number of NP candidates) between each AZP and its counterpart antecedent; **AveLen** means the average length of sentences; and the **AZP/ZP** denotes the percentage of AZPs in ZPs of each source.

From Table 3 we can observe that source TC has the smallest "AntDis" among six sources, where the average distance between each AZP and its counterpart antecedent in TC is only 0.44. As we employ the closest-first search strategy to determine the antecedents, it is therefore easier for the resolver to correctly select the short-distance antecedents than the long-distance ones. In addition, the average length of sentences in source TC is only 10.1, which is far less than in other sources. Considering that the **ZPSNN** models contextual information of a ZP from the beginning to the end of the sentence, a smaller size of context words can avoid a potential exponential decay of the history, which makes it more efficient to represent ZPs. Besides, the average candidates number in source TC is only 4.1 that is far less than in other sources, leading to a smaller search space when selecting antecedents. Under the aforementioned conditions, our approach generates more accurate results, which can be revealed from Table 1. In all three experimental settings, our approach gains extremely high performance with the F-score of 65.2% in source TC, significantly outperforms other sources. On the other side, we can see that AZPs take a great proportion of ZPs in source MZ and NW, where the "AZP/ZP" reaches 77.6% and 80.1%, respectively. Under this situation, less un-anaphoric ZPs will be introduced by applying system AZP identification in these sources. Therefore, the performance of **ZPSNN** in source MZ and NW seems more "stable" with a smaller drop-out in F-score when employing the "System AZP" setting.

To evaluate the effectiveness of attention mechanism, we conduct an experiment on comparing the performance of **ZPSNN** with and without the attention mechanism, as shown in Figure 2.

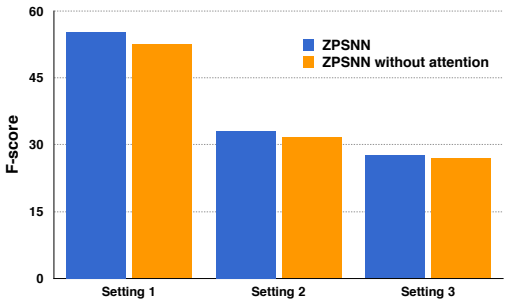

Figure 2: ZP resolution results with and without the attention mechanism.

From Figure 2 we can observe that by removing the attention mechanism from **ZPSNN**, the performance has a decline in all three experimental settings. These results prove that by applying the attention mechanism, our model can more efficiently select the correct antecedent from NP candidates according to the ZP contexts, boosts the performance across the board.

In an attempt to determine the impact of differ-

ent search strategy, which is one of the main issues that can influence the performance of the resolver, we conduct an experiment comparing the best-first search strategy with the closest-first search strategy. To minimum the external influence, we employ experimental setting 1 (gold parse and gold AZP) to carry out the experiment. Table 4 presents the experimental results with different search strategy. We can see that the by employing the best-first search strategy, our resolver only gets 42.2% in overall F-score, which is much lower than that by utilizing the closest-first search strategy. The same results appear in per-sources results, where the closest-first strategy significantly outperforms the best-first strategy in five of the six sources. However, we notice that in source TC, closest-first search strategy beats the closest-first search strategy by 2.9% in F-score, which is idiosyncratic. The reason is that source TC is "Telephone Conversations", where texts are always shorter than in other sources, thus generates fewer NP candidates for a ZP (average candidate number is only 4.1). Admittedly, when applying the best-first strategy, it can make more precious result in a smaller search space. Hence, the best-first search strategy beats the closest-first search strategy in the circumstance with fewer disturbance terms. That suggests that further performance gains might be possible if we can reduce some inaccurate NP candidates before applying the resolver, using a sieve, for instance. Besides, as we utilize the closest-first search strategy to choose antecedents for AZPs, the value of cut-off threshold t becomes very essential. Following (Zhao and Ng, 2007), we set possible values of t from $0.1, 0.15...$ to 0.7. By varying t, the overall F-score changed. Finally, we obtain the best performance at $t = 0.4$[3].

	Best-first			Closest-first		
	R	P	F	R	P	F
Overall	42.2	42.2	42.2	54.7	55.9	**55.3**
NW	30.7	30.7	30.7	45.1	45.1	**45.1**
MZ	19.4	19.4	19.4	44.6	44.6	**44.6**
WB	38.2	38.2	38.2	53.0	53.0	**53.0**
BN	28.7	28.7	28.7	55.5	55.5	**55.5**
BC	50.1	50.1	50.1	59.8	60.5	**60.2**
TC	68.1	68.1	**68.1**	64.3	66.1	65.2

Table 4: Experimental results with difference search strategies.

[3]We tune the parameter t by employing the model trained on 90% of the training data and use the remaining 10% for tuning process. Then we retrain the model on all of the training data before applying it to the test data.

5 Error Analysis

To better evaluate our proposed approach, we perform a case study for experimental results. Our analysis reveals that there are mainly three types of errors, as discussed below.

First, our model may fail when words in NP candidates are not in embedding matrix. The following is an example:

> 威迪奥诺 表示 ϕ 将 把 此 意见 转达 给 联合国 秘书长 。
> Weidiaunuo expressed that ϕ will convey this suggestion to the UN secretary general .

In this case, the correct antecedent of ZP "ϕ" is NP "威迪奥诺/Weidiaunuo" while our model predicts NP "国家/The country" (appears in the preceding contexts) as the result, which is incorrect. We observe that the word "威迪奥诺/Weidiaunuo" has an embedding as $Null$, which makes our model incapable of representing the NP, draws incorrect results. Hence, some more efforts can be achieved, as we can present words that are not in our embedding matrix with appropriate expressions.

The second error appears when a ZP lies at the beginning of a sentence, our model prefers to choose overt pronouns like "你/you" or "我/I", as antecedents. For example,

> ϕ 为何 成为 羡慕 且 嫉妒 的 目标?
> Why ϕ becomes the focus of so much envy and jealousy ?

In this case, ZP "ϕ" should be resolved to NP "台北市/Taibei City" while our model chooses the NP "我们/We". As many of the sentences in our training data start with an overt pronoun, our model thus prefers pronouns to fill the ZP gap. In addition, **ZPSNN** tries to capture contextual information for a ZP by modeling words appear both in its preceding and following contexts. Unfortunately, for aforementioned ZPs, there are no preceding words. Hence, more sufficient contextual information is required to make the right decision. The problem can be better alleviated if we take more contexts into account, by capturing word sequences across the sentence boundaries, acquiring sentence-level history vectors, which will be exploited by us in future work.

The third type of error appears when a ZP has only long-distance antecedents. As we applying the closest-first search strategy to determine

the correct antecedent among NP candidates, our model has acquired the preference for candidates that are closer to the given ZP. Therefore, once there are a great number of NP candidates between a ZP and its closest antecedent, poor resolution results will appear, as in the following example.

> [一些 政府 官员] ， 遇到 [[群众] 反映 的 问题] ， 不是 想 [办法] 解决 [问题] ， 而 ϕ 首先 想到 的 是 压制。
> When facing [problems [the masses] reflect], [some government officials] do not try to find out [solutions] to solve the [problems], but ϕ suppress first.

As we can see that for the ZP "ϕ", there are four NP candidates ("群众/masses", "群众反映的问题/problems the masses reflect","办法/solutions" and "问题/problems") between "ϕ" and its antecedent "一些政府官员/some government officials". In this case, our model incorrectly resolves the ZP "ϕ" to NP "群众/masses". Although our model can exclude the closest two NP candidates, as for discriminating between "群众/masses" and "一些政府官员/some government officials", our model can not deny any of them thus prefers the closer one. Note, however, that it is easy for a human to resolve "ϕ" to "一些政府官员/some government officials" because "ϕ" presents the subject of the last phrase and the NP "一些政府官员/some government officials" also plays the same role in the sentence. If **ZPSNN** knew that the gap was more likely to be filled with the subject, it might produce more precise results. Therefore, to correctly handle such cases, one may integrate grammar information into our model, to provide hints to the resolver when making choices.

6 Related work

In this section, we give a brief review of previous efforts on Chinese zero pronoun resolution task. Early studies employed heuristic rules to Chinese ZP resolution. (Converse, 2006) proposes a rule-based method to resolve the zero pronouns, by utilizing Hobbs algorithm (Hobbs, 1978) in the CTB documents. More recently, supervised approaches to this task have been vastly explored. (Zhao and Ng, 2007) first present a supervised machine learning approach to the identification and resolution of Chinese zero pronouns. In their study, two categories of features that can be summarized to intra- and inter-sentences are exploited. By employing the J48 decision tree model, these features are integrated into the resolution algorithm. (Kong and Zhou, 2010) develop a novel approach for Chinese ZP resolution, employing context-sensitive convolution tree kernels to capture syntactic information. In their framework, there are three sub-processes: zero anaphora detection, anaphor determination, and antecedent identification, all of which are tree-kernel based. (Chen and Ng, 2013) further extend (Zhao and Ng, 2007)'s study by considering more types of novel features. Moreover, they exploit co-reference links between zero pronouns and antecedents. In recent time, (Chen and Ng, 2014) develop an unsupervised language-independent approach. They first recover each zero pronouns into ten overt pronouns and then apply a ranking model to rank the candidate antecedents. They also utilize the integer linear programming to enhance the performance of their ranking model. (Chen and Ng, 2015) propose an end-to-end unsupervised probabilistic model for Chinese ZP resolution, using a salience model to capture discourse information. Experimental results on the OntoNotes 5.0 corpus show that their approach achieves the state-of-the-art performance.

7 Conclusion

In this study, we investigate a novel zero pronoun-specific neural network (**ZPSNN**) for Chinese zero pronoun resolution, exploiting a learning algorithm without extensive feature engineering, meanwhile effectively modeling contextual information to reveal semantic relatedness of the ZP and its NP candidates. Experimental results on the OntoNotes 5.0 corpus show that our proposed approach outperforms the state-of-the-art methods significantly. The future work will be carried out on three aspects. As we have shown that the **ZPSNN** are effective in Chinese zero pronoun resolution, we plan to (1) improve the performance of **ZPSNN** by exploring more efficient word embeddings, which are basic inputs of the neural network; (2) integrate grammar information that plays an important role in predicting the antecedents; (3) capture word sequences across the sentence boundaries, which acquires sentence-level history vectors that can better reveal the semantic information for ZPs.

References

[Bengio et al.2003] Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

[Bengio2013] Yoshua Bengio. 2013. Deep learning of representations: Looking forward. In *Statistical Language and Speech Processing*, pages 1–37. Springer.

[Chen and Ng2013] Chen Chen and Vincent Ng. 2013. Chinese zero pronoun resolution: Some recent advances. In *EMNLP*, pages 1360–1365.

[Chen and Ng2014] Chen Chen and Vincent Ng. 2014. Chinese zero pronoun resolution: An unsupervised approach combining ranking and integer linear programming. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[Chen and Ng2015] Chen Chen and Vincent Ng. 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 ACL and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, page 320.

[Converse2006] Susan P Converse. 2006. Pronominal anaphora resolution in chinese.

[Elman1991] Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2-3):195–225.

[Ferrández and Peral2000] Antonio Ferrández and Jesús Peral. 2000. A computational approach to zero-pronouns in spanish. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 166–172. Association for Computational Linguistics.

[Hermann et al.2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1684–1692. Curran Associates, Inc.

[Hobbs1978] Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Iida et al.2006] Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2006. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 625–632. Association for Computational Linguistics.

[Kim2000] Young-Joo Kim. 2000. Subject/object drop in the acquisition of korean: A cross-linguistic comparison. *Journal of East Asian Linguistics*, 9(4):325–351.

[Kong and Zhou2010] Fang Kong and Guodong Zhou. 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*, pages 882–891. Association for Computational Linguistics.

[Lappin and Leass1994] Shalom Lappin and Herbert J Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

[Mikolov et al.2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

[Mitkov and others1995] Ruslan Mitkov et al. 1995. Anaphora resolution in machine translation. In *Proceedings of the Sixth International conference on Theoretical and Methodological issues in Machine Translation*. Citeseer.

[Ng2007] Vincent Ng. 2007. Semantic class induction and coreference resolution. In *AcL*, pages 536–543.

[Rocktäschel et al.2015] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

[Rush et al.2015] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

[Sasano and Kurohashi2011] Ryohei Sasano and Sadao Kurohashi. 2011. A discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In *IJCNLP*, pages 758–766.

[Vicedo and Ferrández2000] José L Vicedo and Antonio Ferrández. 2000. Importance of pronominal anaphora resolution in question answering systems. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 555–562. Association for Computational Linguistics.

[Zhao and Ng2007] Shanheng Zhao and Hwee Tou Ng. 2007. Identification and resolution of chinese zero pronouns: A machine learning approach. In *EMNLP-CoNLL*, volume 2007, pages 541–550.