

# Chinese Zero Pronoun Resolution: Some Recent Advances

Chen Chen and Vincent Ng  
Human Language Technology Research Institute  
University of Texas at Dallas  
Richardson, TX 75083-0688  
{yzcchen, vince}@hlt.utdallas.edu

## Abstract

We extend Zhao and Ng's (2007) Chinese anaphoric zero pronoun resolver by (1) using a richer set of features and (2) exploiting the coreference links between zero pronouns during resolution. Results on OntoNotes show that our approach significantly outperforms two state-of-the-art anaphoric zero pronoun resolvers. To our knowledge, this is the first work to report results obtained by an *end-to-end* Chinese zero pronoun resolver.

## 1 Introduction

A zero pronoun (ZP) is a gap in a sentence that is found when a phonetically null form is used to refer to a real-world entity. An anaphoric zero pronoun (AZP) is a ZP that corefers with one or more preceding noun phrases (NPs) in the associated text. Unlike overt pronouns, ZPs lack grammatical attributes that are useful for overt pronoun resolution such as number and gender. This makes ZP resolution more challenging than overt pronoun resolution.

We aim to improve the state of the art in Chinese AZP resolution by proposing two extensions. First, while previous approaches to this task have primarily focused on employing positional and syntactic features (e.g., Zhao and Ng (2007) [Z&N], Kong and Zhou (2010) [K&Z]), we exploit a richer set of features for capturing the context of an AZP and its candidate antecedents. Second, to alleviate the difficulty of resolving an AZP to an antecedent far away from it, we break down the process into smaller, intermediate steps, where we allow coreference links between AZPs to be established.

We apply our two extensions to a state-of-the-art Chinese AZP resolver proposed by Z&N and eval-

uate the resulting resolver on the OntoNotes corpus. Experimental results show that this resolver significantly outperforms both Z&N's resolver and another state-of-the-art resolver proposed by K&Z. It is worth noting that while previous work on Chinese ZP resolution has reported results obtained via gold information (e.g., using gold AZPs and extracting candidate antecedents and other features from gold syntactic parse trees), this is the first work to report the results of an *end-to-end* Chinese ZP resolver.

The rest of this paper is organized as follows. Section 2 describes the two baseline AZP resolvers. Sections 3 and 4 discuss our two extensions. We present our evaluation results in Section 5 and our conclusions in Section 6.

## 2 Baseline AZP Resolution Systems

An AZP resolution algorithm takes as input a set of AZPs produced by an AZP identification system. Below we first describe the AZP identifier we employ, followed by our two baseline AZP resolvers.

### 2.1 Anaphoric Zero Pronoun Identification

We employ two steps to *identify* AZPs. In the *extraction* step, we heuristically extract candidate ZPs. In the *classification* step, we train a classifier to distinguish AZPs from non-AZPs.

To implement the extraction step, we use Z&N's and K&Z's observation: ZPs can only occur before a VP node in a syntactic parse tree. However, according to K&Z, ZPs do not need to be extracted from every VP: if a VP node occurs in a coordinate structure or is modified by an adverbial node, then only its parent VP node needs to be considered. We extract ZPs from all VPs that satisfy the above constraints.

Syntactic features (13)	whether $z$ is the first gap in an IP clause; whether $z$ is the first gap in a subject-less IP clause, and if so, $\text{POS}(w_1)$ ; whether $\text{POS}(w_1)$ is NT; whether $t_1$ is a verb that appears in a NP or VP; whether $P_l$ is a NP node; whether $P_r$ is a VP node; the phrasal label of the parent of the node containing $\text{POS}(t_1)$ ; whether V has a NP, VP or CP ancestor; whether C is a VP node; whether there is a VP node whose parent is an IP node in the path from $t_1$ to C.
Lexical features (13)	the words surrounding $z$ and/or their POS tags, including $w_1, w_{-1}, \text{POS}(w_1), \text{POS}(w_{-1})+\text{POS}(w_1), \text{POS}(w_1)+\text{POS}(w_2), \text{POS}(w_{-2})+\text{POS}(w_{-1}), \text{POS}(w_1)+\text{POS}(w_2)+\text{POS}(w_3), \text{POS}(w_{-1})+w_1,$ and $w_{-1}+\text{POS}(w_1)$ ; whether $w_1$ is a transitive verb, an intransitive verb or a preposition; whether $w_{-1}$ is a transitive verb without an object.
Other features (6)	whether $z$ is the first gap in a sentence; whether $z$ is in the headline of the text; the type of the clause in which $z$ appears; the grammatical role of $z$ ; whether $w_{-1}$ is a punctuation; whether $w_{-1}$ is a comma.

Table 1: Features for AZP identification.  $z$  is a zero pronoun. V is the VP node following  $z$ .  $w_i$  is the  $i$ th word to the right of  $z$  (if  $i$  is positive) or the  $i$ th word to the left of  $z$  (if  $i$  is negative). C is lowest common ancestor of  $w_{-1}$  and  $w_1$ .  $P_l$  and  $P_r$  are the child nodes of C that are the ancestors of  $w_{-1}$  and  $w_1$  respectively.

Features between $a$ and $z$ (4)	the sentence distance between $a$ and $z$ ; the segment distance between $a$ and $z$ , where segments are separated by punctuations; whether $a$ is the closest NP to $z$ ; whether $a$ and $z$ are siblings in the associated parse tree.
Features on $a$ (12)	whether $a$ has an ancestor NP, and if so, whether this NP is a descendent of $a$ 's lowest ancestor IP; whether $a$ has an ancestor VP, and if so, whether this VP is a descendent of $a$ 's lowest ancestor IP; whether $a$ has an ancestor CP; the grammatical role of $a$ ; the clause type in which $a$ appears; whether $a$ is an adverbial NP, a temporal NP, a pronoun or a named entity; whether $a$ is in the headline of the text.
Features on $z$ (10)	whether V has an ancestor NP, and if so, whether this NP node is a descendent of V's lowest ancestor IP; whether V has an ancestor VP, and if so, whether this VP is a descendent of V's lowest ancestor IP; whether V has an ancestor CP; the grammatical role of $z$ ; the type of the clause in which V appears; whether $z$ is the first or last ZP of the sentence; whether $z$ is in the headline of the text.

Table 2: Features for AZP resolution in the Zhao and Ng (2007) baseline system.  $z$  is a zero pronoun.  $a$  is a candidate antecedent of  $z$ . V is the VP node following  $z$  in the parse tree.

To implement the classification step, we train a classifier using  $\text{SVM}^{\text{light}}$  (Joachims, 1999) to distinguish AZPs from non-AZPs. We employ 32 features, 13 of which were proposed by Z&N and 19 of which were proposed by Yang and Xue (2010). A brief description of these features can be found in Table 1.

## 2.2 Two Baseline AZP Resolvers

**The Zhao and Ng (2007) [Z&N] baseline.** In our implementation of the Z&N baseline, we use  $\text{SVM}^{\text{light}}$  to train a mention-pair model for determining whether an AZP  $z$  and a candidate antecedent of  $z$  are coreferent. We consider all NPs preceding  $z$  that do not have the same head as its parent NP in the parse tree to be  $z$ 's candidate antecedents. We use Soon et al.'s (2001) method to create training instances: we create a positive instance between an AZP,  $z$ , and its closest overt antecedent, and we create a negative instance between  $z$  and each of the

intervening candidates. Each instance is represented by the 26 features employed by Z&N. A brief description of these features can be found in Table 2. During testing, we adopt the *closest-first* resolution strategy, resolving an AZP to the closest candidate antecedent that is classified as coreferent with it.<sup>1</sup>

**The Kong and Zhou (2010) [K&Z] baseline.** K&Z employ a tree kernel-based approach to AZP resolution. Like Z&N, K&Z (1) train a mention-pair model for determining whether an AZP  $z$  and a candidate antecedent of  $z$  are coreferent, (2) use Soon et al.'s method to create training instances, and (3) resolve an AZP to its closest coreferent candidate antecedent. Unlike Z&N, however, K&Z use the  $\text{SVM}^{\text{light-TK}}$  learning algorithm (Moschitti,

<sup>1</sup>When resolving a *gold* AZP  $z$ , if none of the preceding candidate antecedents is classified as coreferent with it, we resolve it to the candidate that has the highest coreference likelihood with it. Here, we employ the signed distance from the SVM hyperplane to measure the coreference likelihood.

2006) to train their model, employing a parse subtree known as a dynamic expansion tree (Zhou et al., 2008) as a structured feature to represent an instance.

### 3 Extension 1: Novel Features

We propose three kinds of features to better capture the context of an AZP, as described below.

**Antecedent compatibility.** AZPs are omitted subjects that precede VP nodes in a sentence's parse tree. From the VP node, we can extract its head verb ( $Pred_z$ ) and the head of its object NP ( $Obj$ ), if any. Note that  $Pred_z$  and  $Obj$  contain important contextual information for an AZP.

Next, observe that if a NP is coreferent with an AZP, it should be able to fill the AZP's gap and be compatible with the gap's context. Consider the following example:

E1: 他们在试那个服务。那么啊就是 \*pro\* 希望到九月我们的旅客来的时候。

(They are trying that service. That means \*pro\* hope that our visitors can try it when they come in September.)

The head of the VP following \*pro\* is 希望 (hope). There are two candidate antecedents, 他们 (They) and 那个服务 (that service). If we try using them to fill this AZP's gap, we know based on selectional preferences that 他们希望 (They hope) makes more sense than 那个服务希望 (that service hope). We supply the AZP resolver with the following information to help it make these decisions. First, we find the head word of each candidate antecedent,  $Head_c$ . Then we form two strings,  $Head_c + Pred_z$  and  $Head_c + Pred_z + Obj$  (if the object of the VP is present). Finally, we employ them as binary lexical features, setting their feature values to 1 if and only if they can be extracted from the instance under consideration. The training data can be used to determine which of these features are useful.<sup>2</sup>

**Narrative event chains.** A narrative event chain is a partially ordered set of events related by a common protagonist (Chambers and Jurafsky, 2008). For example, we can infer from the chain "borrow-s invest-s spend-s lend-s" that a person who borrows (pre-

<sup>2</sup>We tried to apply Kehler et al.'s (2004) and Yang et al.'s (2005) methods to learn Chinese selectional preferences from unlabeled data, but without success.

sumably money) can invest it, spend it, or lend it to other people.<sup>3</sup> Consider the following example:

E2: 国家给钱了, \*pro\* 提供这部分的钱都是自己所里挣的。

(The country gives our department money, but all \*pro\* provides is exactly what we worked for.)

In E2, \*pro\* is coreferent with 国家 (The country), and the presence of the narrative event chain 给 – 提供 (gives–provides) suggests that the subjects of the two events are likely to be coreferent.

However, given the unavailability of induced or hand-crafted narrative chains in Chinese<sup>4</sup>, we make the simplifying assumption that two verbs form a lexical chain if they are lexically identical.<sup>5</sup> We create two features to exploit narrative event chains for a candidate NP,  $c$ , if it serves as a subject or object. Specifically, let the verb governing  $c$  be  $Pred_c$ . The first feature, which encodes whether narrative chains are present, has three possible values: 0 if  $Pred_c$  and  $Pred_z$  are not the same; 1 if  $Pred_c$  and  $Pred_z$  are the same and  $c$  is a subject; and 2 if  $Pred_c$  and  $Pred_z$  are the same and  $c$  is an object. The second feature is a binary lexical feature,  $Pred_c + Pred_z + Subject/Object$ ; its value is 1 if and only if  $Pred_c$ ,  $Pred_z$ , and  $Subject/Object$  can be found in the associated instance, where  $Subject/Object$  denotes the grammatical role of  $c$ .

**Final punctuation hint.** We observe that the punctuation ( $Punc$ ) at the end of a sentence where an AZP occurs also provides contextual information, especially in conversation documents. In conversations, if a sentence containing an AZP ends with a

<sup>3</sup>"-s" denotes the fact that the protagonist serves as the grammatical subject in these events.

<sup>4</sup>We tried to construct narrative chains for Chinese using both learning-based and dictionary-based methods. Specifically, we induced narrative chains using Chambers and Jurafsky's (2008) method, but were not successful owing to the lack of an accurate Chinese coreference resolver. In addition, we constructed narrative chains using both lexically identical verbs and the synonyms obtained from a WordNet-like Chinese resource called Tongyicilin, but they did not help improve resolution performance.

<sup>5</sup>Experiments on the training data show that if an AZP and a candidate antecedent are subjects of (different occurrences of) the same verb, then the probability that the candidate antecedent is coreferent with the AZP is 0.703. This result suggests that our assumption, though somewhat simplistic, is useful as far as AZP resolution is concerned.

---

A: 她现在生活怎么样?  
(A: How is **her** life now? )  
B: *\*pro<sub>1</sub>\** 对生活, 就是很朴素, 很简单。  
(B: *\*pro<sub>1</sub>\** attitude toward life is plain and simple.)  
A: 嗯。  
(A: Yes.)  
A: *\*pro<sub>2</sub>\** 是在北京还是在美国?  
(A: *\*pro<sub>2</sub>\** is living in Beijing or the USA?)  
B: *\*pro<sub>3</sub>\** 在美国。  
(B: *\*pro<sub>3</sub>\** is living in the USA.)

---

Figure 1: An illustrative example.

question mark, the mention this AZP refers to is less likely to be the speaker himself<sup>6</sup>, as illustrated in the following example:

E3: 冬天 *\*pro\** 冷吗?  
(Are *\*pro\** cold in the winter?)

Here, *\*pro\** refers to the person the speaker talks with. To capture this information, we create a binary lexical feature, *Head<sub>c</sub>+Punc*, whose value is 1 if and only if *Head<sub>c</sub>* and *Punc* appear in the instance under consideration.

## 4 Extension 2: Zero Pronoun Links

### 4.1 Motivation

Like an overt pronoun, a ZP whose closest overt antecedent is far away from it is harder to resolve than one that has a nearby overt antecedent. However, a corpus study of our training data reveals that only 55.2% of the AZPs appear in the same sentence as their closest overt antecedent, and 22.7% of the AZPs appear two or more sentences away from their closest overt antecedent.

Fortunately, we found that some of the difficult-to-resolve AZPs (i.e., AZPs whose closest overt antecedents are far away from them) are coreferential with nearby ZPs. Figure 1, which consists of a set of sentences from a conversation, illustrates this phenomenon. There are three AZPs (denoted by *\*pro<sub>i</sub>\**, where  $1 \leq i \leq 3$ ), all of which refer to the overt pronoun 她 (She) in the first sentence. In this example, it is fairly easy to resolve *\*pro<sub>1</sub>\** correctly,

<sup>6</sup>One may wonder whether we can similarly identify constraints on the antecedents of a ZP from clause conjunctions. Our preliminary analysis suggests that the answer is no.

	Training	Test
Documents	1,391	172
Sentences	36,487	6,083
Words	756,063	110,034
ZPs	23,065	3,658
AZPs	12,111	1,713

Table 3: Statistics on the training and test sets.

since its antecedent is the subject of previous sentence. However, *\*pro<sub>3</sub>\** and its closest overt antecedent 她 (She) are four sentences apart. Together with the fact that there are many intervening candidate antecedents, it is not easy for a resolver to correctly resolve *\*pro<sub>3</sub>\**.

To facilitate the resolution of *\*pro<sub>3</sub>\** and difficult-to-resolve AZPs in general, we propose the following idea. We allow an AZP resolver to (1) establish coreferent links between two consecutive ZPs (i.e., *\*pro<sub>1</sub>\**–*\*pro<sub>2</sub>\** and *\*pro<sub>2</sub>\**–*\*pro<sub>3</sub>\** in our example), which are presumably easy to establish because the two AZPs involved are close to each other; and then (2) treat them as bridges and infer that *\*pro<sub>3</sub>\**'s overt antecedent is 她 (She).

### 4.2 Modified Resolution Algorithm

We implement the aforementioned idea by modifying the AZP resolver as follows. When we resolve an AZP  $z$  during testing, we augment the set of candidate antecedents for  $z$  with the set of AZPs preceding  $z$ . Since we have only specified how to compute features for instances composed of an AZP and an overt candidate antecedent thus far (see Section 2.2), the question, then, is: how can we compute features for instances composed of two AZPs?

To answer this question, we first note that the AZPs in a test text are resolved in a left-to-right manner. Hence, by the time we resolve an AZP  $z$ , all the AZPs preceding  $z$  have been resolved. Hence, when we create a test instance  $i$  between  $z$  and one of the preceding AZPs (say  $y$ ), we create  $i$  as if the gap  $y$  was filled with the smallest tree embedding the NP to which  $y$  was resolved.

By allowing coreference links between (presumably nearby) ZPs to be established, we can reason over the resulting coreference links, treating them as bridges that can help us find an overt antecedent that is far away from an AZP.

System Variation	Gold AZP			System AZP			System AZP		
	Gold Parse Tree			Gold Parse Tree			System Parse Tree		
	R	P	F	R	P	F	R	P	F
K&Z Baseline System	38.0	38.0	38.0	17.7	22.4	19.8	10.6	13.6	11.9
Z&N Baseline System	41.5	41.5	41.5	22.4	24.4	23.3	12.7	14.2	13.4
Z&N Baseline + Contextual Features	46.2	46.2	46.2	25.2	27.5	26.3	14.4	16.1	15.2
Z&N Baseline + Zero Pronoun Links	42.7	42.7	42.7	22.5	24.6	23.5	13.2	14.8	13.9
Full System	<b>47.7</b>	<b>47.7</b>	<b>47.7</b>	<b>25.3</b>	<b>27.6</b>	<b>26.4</b>	<b>14.9</b>	<b>16.7</b>	<b>15.7</b>

Table 4: Resolution results on the test set.

## 5 Evaluation

### 5.1 Experimental Setup

**Dataset.** For evaluation, we employ the portion of the OntoNotes 4.0 corpus that was used in the official CoNLL-2012 shared task. The shared task dataset is composed of a training set, a development set, and a test set. Since only the training set and the development set are annotated with ZPs, we use the training set for classifier training and reserve the development set for testing purposes. Statistics on the datasets are shown in Table 3. In these datasets, a ZP is marked as *\*pro\**. We consider a ZP anaphoric if it is coreferential with a preceding ZP or overt NP.

**Evaluation measures.** We express the results of both AZP identification and AZP resolution in terms of recall (R), precision (P) and F-score (F).

### 5.2 Results and Discussion

The three major columns of Table 4 show the results obtained in three settings, which differ in terms of whether gold/system AZPs and manually/automatically constructed parse trees are used to extract candidate antecedents and features.

In the first setting, the resolvers are provided with gold AZPs and gold parse trees. Results are shown in column 1. As we can see, the Z&N baseline significantly outperforms the K&Z baseline by 3.5% in F-score.<sup>7</sup> Adding the contextual features, the ZP links, and both extensions to Z&N increase its F-score significantly by 4.7%, 1.2% and 6.2%, respectively.

In the next two settings, the resolvers operate on the *system* AZPs provided by the AZP identification component. When gold parse trees are employed, the recall, precision and F-score of AZP identification are 50.6%, 55.1% and 52.8% respectively. Column 2 shows the results of the resolvers obtained

when these automatically identified AZPs are used. As we can see, Z&N again significantly outperforms K&Z by 3.5% in F-score. Adding the contextual features, the ZP links, and both extensions to Z&N increase its F-score by 3.0%, 0.2% and 3.1%, respectively. The system with contextual features and the full system both yield results that are significantly better than those of the Z&N baseline. A closer examination of the results reveals why the ZP links are not effective in improving performance: when employing system AZPs, many erroneous ZP links were introduced to the system.

Column 3 shows the results of the resolvers when we employ system AZPs and the automatically generated parse trees provided by the CoNLL-2012 shared task organizers to compute candidate antecedents and features. Hence, these are *end-to-end* ZP resolution results. To our knowledge, these are the first reported results on end-to-end Chinese ZP resolution. Using automatic parse trees, the performance on AZP identification drops to 30.8% (R), 34.4% (P) and 32.5% (F). In this setting, Z&N still outperforms K&Z significantly, though by a smaller margin when compared to the previous settings. Incorporating the contextual features, the ZP links, and both extensions increase the F-score by 1.8%, 0.5% and 2.3%, respectively. The system with contextual features and the full system both yield results that are significantly better than those of the Z&N baseline.

## 6 Conclusions

We proposed two extensions to a state-of-the-art Chinese AZP resolver proposed by Zhao and Ng (2007). Experimental results on the OntoNotes dataset showed that the resulting resolver significantly improved both Zhao and Ng's and Kong and Zhou's (2010) resolvers, regardless of whether gold or system AZPs and syntactic parse trees are used.

<sup>7</sup>All significance tests are paired *t*-tests, with  $p < 0.05$ .

## Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

## References

- Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 787--797.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Scholkopf and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 44--56. MIT Press.
- Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. Competitive self-trained pronoun interpretation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 33--36.
- 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.
- Alessandro Moschitti. 2006. Making tree kernels practical for natural language processing. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113--120.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521--544.
- Yaqin Yang and Nianwen Xue. 2010. Chasing the ghost: recovering empty categories in the Chinese Treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1382--1390.
- Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 165--172.
- Shanheng Zhao and Hwee Tou Ng. 2007. Identification and resolution of Chinese zero pronouns: A machine learning approach. In *Proceedings of the 2007 Joint Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning*, pages 541--550.
- GuoDong Zhou, Fang Kong, and Qiaoming Zhu. 2008. Context-sensitive convolution tree kernel for pronoun resolution. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 25--31.