# Chinese Zero Pronoun Resolution: An Unsupervised Probabilistic Model Rivaling Supervised Resolvers

**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

State-of-the-art Chinese zero pronoun resolution systems are supervised, thus relying on training data containing manually resolved zero pronouns. To eliminate the reliance on annotated data, we present a generative model for unsupervised Chinese zero pronoun resolution. At the core of our model is a novel hypothesis: a probabilistic pronoun resolver trained on overt pronouns in an unsupervised manner can be used to resolve zero pronouns. Experiments demonstrate that our unsupervised model rivals its state-of-the-art supervised counterparts in performance when resolving the Chinese zero pronouns in the OntoNotes corpus.

## 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. Below is an example taken from the Chinese TreeBank (CTB), where the ZP (denoted as *pro*) refers to 俄罗斯 (Russia).

[俄罗斯] 作为米洛舍夫维奇一贯的支持者，*pro* 曾经提出调停这场政治危机。

([Russia] is a consistent supporter of Milošević, *pro* has proposed to mediate the political crisis.)

As we can see, 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.

Automatic ZP resolution is typically composed of two steps. The first step, AZP identification, involves extracting ZPs that are anaphoric. The second step, AZP resolution, aims to identify an antecedent of an AZP. State-of-the-art ZP resolvers have tackled both of these steps in a supervised manner, training a classifier for AZP identification and another one for AZP resolution (e.g., Zhao and Ng (2007), Chen and Ng (2013)).

In this paper, we focus on the second task, AZP resolution, designing a model that assumes as input the AZPs in a document and resolves each of them. Note that the task of AZP resolution alone is by no means easy: even when gold-standard AZPs are given, state-of-the-art supervised resolvers can only achieve an F-score of 47.7% for *resolving* Chinese AZPs (Chen and Ng, 2013). For the sake of completeness, we will evaluate our AZP resolution model using both gold-standard AZPs as well as AZPs automatically identified by a rule-based approach that we propose in this paper.

Our contribution lies in the proposal of the first *unsupervised* probabilistic model for AZP resolution that rivals its supervised counterparts in performance when evaluated on the Chinese portion of the OntoNotes 5.0 corpus. Its main advantage is that it does not require training data with manually resolved AZPs. This, together with the fact that its underlying generative process is not language-dependent, enables it to be applied to languages where such annotated data is not readily available. At its core is a novel hypothesis: we can apply a probabilistic pronoun resolution model trained on *overt* pronouns in an *unsupervised* manner to resolve *zero* pronouns. Motivated by Cherry and Bergsma's (2005) and Charniak and Elsner's (2009) work on unsupervised English pronoun resolution, we train our unsupervised resolver on Chinese overt pronouns using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

## 2 Related Work

**Chinese ZP resolution.** Early approaches to Chinese ZP resolution are *rule-based*. Converse (2006) applied Hobbs' algorithm (Hobbs,

*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 763–774,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

1978) to resolve the ZPs in the CTB documents. Yeh and Chen (2007) hand-engineered a set of rules for ZP resolution based on Centering Theory (Grosz et al., 1995).

In contrast, virtually all recent approaches to this task are based on *supervised* learning. Zhao and Ng (2007) are the first to employ a supervised learning approach to Chinese ZP resolution. They trained an AZP resolver by employing syntactic and positional features in combination with a decision tree learner. Unlike Zhao and Ng, Kong and Zhou (2010) employed context-sensitive convolution tree kernels (Zhou et al., 2008) in their resolver to model syntactic information. More recently, we extended Zhao and Ng's feature set with novel features that encode the context surrounding a ZP and its candidate antecedents, and exploited the coreference links between ZPs as bridges to find textually distant antecedents for ZPs (Chen and Ng, 2013).

**ZP resolution for other languages.** There have been rule-based and supervised machine learning approaches for resolving ZPs in other languages. For example, to resolve ZPs in Spanish texts, Ferrández and Peral (2000) proposed a set of handcrafted rules that encode preferences for candidate antecedents. In addition, supervised approaches have been extensively employed to resolve ZPs in Korean (e.g., Han (2006)), Japanese (e.g., Seki et al. (2002), Isozaki and Hirao (2003), Iida et al. (2006; 2007), Imamura et al. (2009), Iida and Poesio (2011), Sasano and Kurohashi (2011)), and Italian (e.g., Iida and Poesio (2011)).

## 3 Chinese Overt Pronouns

Since our approach relies heavily on Chinese *overt* pronouns, in this section we introduce them by describing their four grammatical attributes, namely NUMBER, GENDER, PERSON and ANIMACY. NUMBER has two values, *singular* and *plural*. GENDER has three values, *neuter*, *masculine* and *feminine*. PERSON has three values, *first*, *second* and *third*. Finally, ANIMACY has two values, *animate* and *inanimate*.

We exploit ten personal pronouns that have well-defined grammatical attribute values, namely 你 (singular you), 我 (I), 他 (he), 她 (she), 它 (it), 你们 (plural you), 我们 (we), 他们 (masculine they), 她们 (feminine they), and 它们 (impersonal they). As can be seen in Table 1, each of them can be uniquely identified using these four attributes.

| Pronouns | NUMBER | GENDER | PERSON | ANIMACY |
|---|---|---|---|---|
| 我 (I) | singular | neuter | first | animate |
| 你 (you) | singular | neuter | second | animate |
| 他 (he) | singular | masculine | third | animate |
| 她 (she) | singular | feminine | third | animate |
| 它 (it) | singular | neuter | third | inanimate |
| 你们 (you) | plural | neuter | second | animate |
| 我们 (we) | plural | neuter | first | animate |
| 他们 (they) | plural | masculine | third | animate |
| 她们 (they) | plural | feminine | third | animate |
| 它们 (they) | plural | neuter | third | inanimate |

Table 1: Attribute values of Chinese overt pronouns.

## 4 The Generative Model

### 4.1 Notation

Let $p$ be an overt pronoun in $PR$, the set of the 10 overt pronouns described in Section 3. $C$, the set of candidate antecedents of $p$, contains all and only those maximal or modifier NPs that precede $p$ in the associated text and are at most two sentences away from it.[1] $k$ is the context surrounding $p$ as well as every candidate antecedent $c$ in $C$; $k_c$ is the context surrounding $p$ and candidate antecedent $c$; and $l$ is a binary variable indicating whether $c$ is the correct antecedent of $p$. The set $A = \{Num, Gen, Per, Ani\}$ has four elements, which correspond to NUMBER, GENDER, PERSON and ANIMACY respectively. $a$ is an attribute in $A$. Finally, $p_a$ and $c_a$ are the attribute values of $p$ and $c$ with respect to $a$ respectively.

### 4.2 Training

Our model estimates $P(p, k, c, l)$, the probability of seeing (1) the overt pronoun $p$; (2) the context $k$ surrounding $p$ and its candidate antecedents; (3) a candidate antecedent $c$ of $p$; and (4) whether $c$ is the correct antecedent of $p$. Since we estimate this probability from a raw, unannotated corpus, we are effectively treating $p$, $k$, and $c$ as observed data and $l$ as hidden data.

Owing to the presence of hidden data, we estimate the model parameters using the EM algorithm. Specifically, we use EM to iteratively estimate the model parameters from data in which each overt pronoun is labeled with the probability it corefers with each of its candidate antecedents and apply the resulting model to re-label each overt pronoun with the probability it corefers with each of its candidate antecedents. Below we describe

[1] Only 8% of the overt pronouns in our corpus, the Chinese portion of the OntoNotes 5.0 corpus, do not have any antecedent in the preceding two sentences.

the details of the E-step and the M-step.

### 4.2.1 E-Step

The goal of the E-step is to compute $P(l{=}1|p,k,c)$, the probability that a candidate antecedent $c$ is the correct antecedent of $p$ given context $k$. Assuming that exactly one of the $p$'s candidate antecedents is its correct antecedent, we can rewrite $P(l{=}1|p,k,c)$ as follows:

$$P(l{=}1|p,k,c) = \frac{P(p,k,c,l{=}1)}{\sum_{c' \in C} P(p,k,c',l{=}1)} \quad (1)$$

Applying Chain Rule, we can rewrite $P(p,k,c,l{=}1)$ as follows:

$$\begin{aligned} P(p,k,c,l{=}1) &= P(p|k,c,l{=}1) * P(l{=}1|k,c) \\ &\quad * P(c|k) * P(k) \end{aligned} \quad (2)$$

Next, given $l = 1$ (i.e., $c$ is the antecedent of $p$), we assume that we can generate $p$ from $c$ without looking at the context.[2] Then we represent $p$ using its grammatical attributes $A$. We further assume that $p$'s value with respect to attribute $a \in A$ is independent of the value of each of its remaining attributes given the antecedent's value with respect to $a$. So we can rewrite $P(p|k,c,l{=}1)$ as follows:

$$\begin{aligned} &P(p|k,c,l{=}1) \approx P(p|c,l{=}1) \\ &\approx P(p_{Num}, p_{Gen}, p_{Per}, p_{Ani}|c,l{=}1) \\ &\approx \prod_{a \in A} P(p_a|c_a,l{=}1) \end{aligned} \quad (3)$$

Moreover, we assume that (1) given $p$ and $c$'s context, the probability of $c$ being the antecedent of $p$ is not affected by the context of the other candidate antecedents; and (2) $k_c$ is sufficient for determining whether $c$ is the antecedent of $p$. So,

$$P(l{=}1|k,c) \approx P(l{=}1|k_c,c) \approx P(l{=}1|k_c) \quad (4)$$

Furthermore, we assume that given context $k$, each candidate antecedent of $p$ is generated with equal probability. In other words,

$$P(c|k) \approx P(c'|k) \ \ \forall\, c,c' \in C \quad (5)$$

Given Equations (2), (3), (4) and (5), we can rewrite $P(l{=}1|p,k,c)$ as:

$$\begin{aligned} P(l{=}1|p,k,c) &= \frac{P(p,k,c,l{=}1)}{\sum_{c' \in C} P(p,k,c',l{=}1)} \\ \approx &\frac{\prod_{a \in A} P(p_a|c_a,l{=}1) * P(l{=}1|k_c)}{\sum_{c' \in C} \prod_{a \in A} P(p_a|c'_a,l{=}1) * P(l{=}1|k_{c'})} \end{aligned} \quad (6)$$

As we can see from Equation (6), our model has two groups of parameters, namely $P(p_a|c_a,l{=}1)$ and $P(l{=}1|k_c)$. Since we have four grammatical attributes, $P(p_a|c_a,l{=}1)$ contains four sets of parameters, with one set per attribute. Using Equation (6) and the current parameter estimates, we can compute $P(l{=}1|p,k,c)$.

Two points deserve mention before we describe the M-step. First, we estimate $P(l{=}1|p,k,c)$ from all and only those overt pronouns $p \in PR$ that are surface or deep subjects in their corresponding sentences. This condition is motivated by our observation that 99.56% of the ZPs in our evaluation corpus (i.e., OntoNotes 5.0) are surface or deep subjects. In other words, we impose this condition so that we can focus our efforts on learning a model for resolving overt pronouns that are subjects. This is by no means a limitation of our model: if we were given a corpus in which many ZPs occur as grammatical objects, we could similarly train another model on overt objects. Second, since in the E-step we attempt to probabilistically label every overt pronoun $p$ that satisfies the condition above, our model is effectively making the simplifying assumption that every overt pronoun is anaphoric. This is clearly an overly simplistic assumption. One way to relax this assumption, which we leave as future work, is to first identify those pronouns that are anaphoric and then use EM to estimate the joint probability only from the anaphoric pronouns.

### 4.2.2 M-Step

Given $P(l{=}1|p,k,c)$, the goal of the M-step is to (re)estimate the model parameters, $P(p_a|c_a,l{=}1)$ and $P(l{=}1|k_c)$, using maximum likelihood estimation. Specifically, $P(p_a|c_a,l{=}1)$ is estimated as follows:

$$P(p_a|c_a,l{=}1) = \frac{Count(p_a,c_a,l{=}1) + \theta}{Count(c_a,l{=}1) + \theta * |a|} \quad (7)$$

where $Count(c_a,l{=}1)$ is the expected number of times $c$ has attribute value $c_a$ when it is the antecedent of $p$; $|a|$ is the number of possible values of attribute $a$; $\theta$ is the Laplace smoothing parameter, which we set to 1; and $Count(p_a,c_a,l{=}1)$ is the expected number of times $p$ has attribute value $p_a$ when its antecedent $c$ has attribute value $c_a$. Given attribute values $p'_a$ and $c'_a$, we compute

[2] This assumption is reasonable because it is fairly easy to determine which pronoun can be used to refer to a given NP.

$Count(p'_a, c'_a, l{=}1)$ as follows:

$$Count(p'_a, c'_a, l{=}1) = \sum_{p,c:p_a=p'_a,c_a=c'_a} P(l{=}1|p,k,c) \quad (8)$$

Similarly, $P(l{=}1|k_c)$ is estimated as follows:

$$P(l{=}1|k_c) = \frac{Count(k_c, l{=}1) + \theta}{Count(k_c) + \theta * 2} \quad (9)$$

where $Count(k_c)$ is the number of times $k_c$ appears in the training data, and $Count(k_c, l{=}1)$ is the expected number of times $k_c$ is the context surrounding a pronoun and its antecedent $c$. Given context $k'_c$, we compute $Count(k'_c, l{=}1)$ as follows:

$$Count(k'_c, l{=}1) = \sum_{k:k_c=k'_c} P(l{=}1|p,k,c) \quad (10)$$

To start the induction process, we initialize all parameters with uniform values. Specifically, $P(p_a|c_a, l{=}1)$ is set to $\frac{1}{|a|}$, and $P(l{=}1|k_c)$ is set to $0.5$. Then we iteratively run the E-step and the M-step until convergence.

There are two important questions we have not addressed. First, how can we compute the four attribute values of a candidate antecedent (i.e., $c_a$ for each attribute $a$), which we need to estimate $P(p_a|c_a, l{=}1)$? Second, what features should we use to represent context $k_c$, which we need to estimate $P(l{=}1|k_c)$? We defer the discussion of these questions to Sections 5 and 6.

### 4.3 Inference

After training, we can apply the resulting model to resolve AZPs. Given an AZP $z$, we determine its antecedent as follows:

$$(\hat{c}, \hat{p}) = \underset{c \in C,\ p \in PR}{\arg\max}\ P(l{=}1|p,k,c) \quad (11)$$

where $PR$ is our set of 10 Chinese overt pronouns and $C$ is the set of candidate antecedents of $z$. In other words, we apply Formula (11) to each AZP $z$, searching for the candidate antecedent $c$ and overt pronoun $p$ that maximize $P(l{=}1|p,k,c)$ when $p$ is used to fill the ZP gap left behind by $z$. The $c$ that results in the maximum probability value over all overt pronouns in $PR$ is chosen as the antecedent of $z$. In essence, since the model is trained on overt pronouns but is applied to ZPs, we have to exhaustively fill the ZP's gap under consideration with each of the 10 overt pronouns in $PR$ during inference.

Although we can now apply our generative model to resolve AZPs, the resolution procedure can be improved further. The improvement is motivated by a problem we observed previously (Chen and Ng, 2013): an AZP and its closest antecedent can sometimes be far away from each other, thus making it difficult to correctly resolve the AZP. To address this problem, we employ the following resolution procedure in our experiments. Given a test document, we process its AZPs in a left-to-right manner. As soon as we resolve an AZP to a preceding NP $c$, we fill the corresponding AZP's gap with $c$. Hence, when we process an AZP $z$, all of its preceding AZPs in the associated text have been resolved, with their gaps filled by the NPs they are resolved to. To resolve $z$, we create test instances between $z$ and its candidate antecedents in the same way as described before. The only difference is that the set of candidate antecedents of $z$ may now include those NPs that are used to fill the gaps of the AZPs resolved so far. In other words, this incremental resolution procedure may increase the number of candidate antecedents of each AZP $z$. Some of these additional candidate antecedents are closer to $z$ than the original candidate antecedents, thereby facilitating the resolution of $z$. If the model resolves $z$ to the additional candidate antecedent that fills the gap left behind by, say, AZP $z'$, we postprocess the output by resolving $z$ to the NP that $z'$ is resolved to.[3]

## 5 Attributes of Candidate Antecedents

In this section, we describe how we determine the four grammatical attribute values (NUMBER, GENDER, PERSON and ANIMACY) of a candidate antecedent $c$, as they are used to represent $c$ when estimating $P(p_a|c_a, l{=}1)$ for each attribute $a$.

### 5.1 ANIMACY

We determine the ANIMACY of a candidate antecedent $c$ heuristically. Specifically, we first check the NP type of $c$. If $c$ is a pronoun, we look up its ANIMACY in Table 1. If $c$ is a named entity, there are two cases to consider: if $c$ is a *person*[4], we label it as *animate*; otherwise, we label it as *inanimate*. If $c$ is a common noun, we look up the ANIMACY of its head noun in an automatically

[3] This postprocessing step is needed because the additional candidate antecedents are only gap fillers.

[4] A detailed description of our named entity recognizer can be found in Chen and Ng (2014).

constructed word list $WL$. If the head noun is not in $WL$, we set its ANIMACY to *unknown*.

Our method for constructing $WL$ is motivated by an observation of measure words in Chinese: some of them only modify *inanimate* nouns while others only modify *animate* nouns. For example, the nouns modified by the measure word 张 are always *inanimate*, as in 一张纸 (one piece of paper). On the other hand, the nouns modified by the measure word 位 are always *animate*, as in 一位工人 (one worker).

Given this observation, we first define two lists, $M_{ani}$ and $M_{inani}$. $M_{ani}$ is a list of measure words that can only modify *animate* nouns. $M_{inani}$ is a list of measure words that can only modify *inanimate* nouns.[5] There exists a special measure word in Chinese, 个, which can be used to modify most of the common nouns regardless of their ANIMACY. As a result, we remove 个 from both lists. After constructing $M_{ani}$ and $M_{inani}$, we (1) parse the Chinese Gigaword corpus (Parker et al., 2009), which contains 4,370,600 documents, using an efficient dependency parser, ctbparser[6] (Qian et al., 2010), and then (2) collect all pairs of words $(m, n)$, where $m$ is a measure word, $n$ is a common noun, and there is a NMOD dependency relation between $m$ and $n$. Finally, we determine the ANIMACY of a given common noun $n$ as follows. First, we retrieve all of the pairs containing $n$. Then, we sum over all occurrences of $m$ in $M_{ani}$ (call the sum $C_{ani}$), as well as all occurrences of $m$ in $M_{inani}$ (call the sum $C_{inani}$). If $C_{ani} > C_{inani}$, we label this common noun as *animate*; otherwise, we label it as *inanimate*.

Table 2 shows the learned values of $P(p_{Ani}|c_{Ani}, l{=}1)$. These results are consistent with our intuition: an animate (inanimate) pronoun is more likely to be generated from an animate (inanimate) antecedent than from an inanimate (animate) antecedent. Note that animate pronouns are more likely to be generated than inanimate pronouns regardless of the antecedent's ANIMACY. This can be attributed to the fact that 94.6% of the pronouns in our corpus are animate.

## 5.2 GENDER

We determine the GENDER of a candidate antecedent $c$ as follows. If $c$ is a pronoun, we look up its GENDER in Table 1. Otherwise, we determine its GENDER based on its ANIMACY. Specifically, if $c$ is *inanimate*, we set its GENDER to *neuter*. Otherwise, we determine its gender by looking up a gender word list constructed by Bergsma and Lin's (2006) approach. If the word is not in the list, we set its GENDER to *masculine* by default.

| Pronoun<br>Antecedent | animate | inanimate |
|---|---|---|
| animate | 0.999 | 0.001 |
| inanimate | 0.858 | 0.142 |
| unknown | 0.945 | 0.055 |

Table 2: Learned values of $P(p_{Ani}|c_{Ani}, l{=}1)$.

Next, we describe how the aforementioned gender word list is constructed. Following Bergsma and Lin (2006), we define a *dependency path* as the sequence of non-terminal nodes and dependency labels between two potentially coreferent entities in a dependency parse tree. From the parsed Chinese Gigaword corpus, we first collect every dependency path that connects two pronouns. For each path $P$ collected, we compute $CL(P)$, the coreference likelihood of $P$, as follows:

$$CL(P) = \frac{N_I(P)}{N_I(P) + N_D(P)} \qquad (12)$$

where $N_I(P)$ is the number of times $P$ connects two identical pronouns, and $N_D(P)$ is the number of times it connects two different pronouns. Assuming that two identical pronouns in a sentence are coreferent (Bergsma and Lin, 2006), we can see that the larger a path's $CL$ value is, the more likely it is that the two NPs it connects are coreferent. To ensure that we have dependency paths that are strongly indicative of coreference relations, we consider a dependency path $P$ a *coreferent path* if and only if $CL(P) > 0.8$.

Given these coreferent paths, we can compute the GENDER of a noun $n$ as follows. First, we compute (1) $N_M(n)$, the number of coreferent paths connecting $n$ with a *masculine* pronoun; and (2) $N_F(n)$, the number of coreferent paths connecting $n$ with a *feminine* pronoun. Then, if $N_F(n) > N_M(n)$, we set $n$'s gender to *feminine*; otherwise, we set it to *masculine*.

Table 3 shows the learned values of $P(p_{Gen}|c_{Gen}, l{=}1)$. These results are consistent with our intuition: a pronoun is a lot more likely to be generated from an antecedent with the same GENDER than one with a different GENDER.

[5] We create these two lists with the help of this page: `http://chinesenotes.com/ref_measure_words.htm`

[6] `http://code.google.com/p/ctbparser/`

| Pronoun<br>Antecedent | neuter | feminine | masculine |
|---|---|---|---|
| neuter | 0.864 | 0.018 | 0.117 |
| feminine | 0.065 | 0.930 | 0.005 |
| masculine | 0.130 | 0.041 | 0.828 |

Table 3: Learned values of $P(p_{Gen}|c_{Gen}, l{=}1)$.

| Pronoun<br>Antecedent | singular | plural |
|---|---|---|
| singular | 0.861 | 0.139 |
| plural | 0.26 | 0.74 |

Table 4: Learned values of $P(p_{Num}|c_{Num}, l{=}1)$.

### 5.3 NUMBER

When computing the NUMBER of a candidate antecedent in English, Charniak and Elsner (2009) rely on part-of-speech information. For example, NN and NNP denote singular nouns, whereas NNS and NNPS denote plural nouns. However, Chinese part-of-speech tags do not provide such information. Hence, we need a different method for finding the NUMBER of a candidate antecedent $c$ in Chinese. If $c$ is a pronoun, we look up its NUMBER in Table 1. If $c$ is a named entity, its NUMBER is *singular*. If $c$ is a common noun, we infer its NUMBER from its string: if the string ends with 们 or is modified by a quantity word (e.g., 一些, 许多), $c$ is *plural*; otherwise, $c$ is *singular*.

Table 4 shows the learned values of $P(p_{Num}|c_{Num}, l{=}1)$. These results are consistent with our intuition: a pronoun is more likely to be generated from an antecedent with the same NUMBER than one with a different NUMBER.

### 5.4 PERSON

Finally, we compute the PERSON of a candidate antecedent $c$. Similar to Charniak and Elsner (2009), we set 我 (I) and 我们 (we) to *first* person, 你 (singular you) and 你们 (plural you) to *second* person, and everything else to *third* person. We estimate two sets of probabilities $P(p_{Per}|c_{Per}, l{=}1)$, one where $p$ and $c$ are from the same speaker, and the other where they are from different speakers.[7] This is based on our observation that $P(p_{Per}|c_{Per}, l{=}1)$ could be very different in these two cases.

| Pronoun<br>Antecedent | first | second | third |
|---|---|---|---|
| first | 0.856 | 0.119 | 0.025 |
| second | 0.219 | 0.766 | 0.016 |
| third | 0.289 | 0.077 | 0.634 |

Table 5: Learned values of $P(p_{Per}|c_{Per}, l{=}1)$ (same speaker).

| Pronoun<br>Antecedent | first | second | third |
|---|---|---|---|
| first | 0.417 | 0.525 | 0.057 |
| second | 0.75 | 0.23 | 0.02 |
| third | 0.437 | 0.229 | 0.334 |

Table 6: Learned values of $P(p_{Per}|c_{Per}, l{=}1)$ (different speakers).

Tables 5 and 6 show the learned values of these two sets of probabilities. These results are consistent with our intuition. In the same-speaker case, a pronoun is a lot more likely to be generated from an antecedent with the same speaker than one with a different speaker. In the different-speaker case, a first (second) person pronoun is most likely to be generated from a second (first) person pronoun.

## 6 Context Features

To fully specify our model, we need to describe how to represent $k_c$, which is needed to compute $P(l{=}1|k_c)$. Recall that $k_c$ encodes the context surrounding candidate antecedent $c$ and the associated pronoun $p$. As described below, we represent $k_c$ using eight features, some of which are motivated by previous work on supervised AZP resolution (e.g., Zhao and Ng (2007), Chen and Ng (2013)). Note that (1) all but feature 1 are computed based on syntactic parse trees, and (2) features 2, 3, 6, and 8 are ternary-valued features.

1. the sentence distance between $c$ and $p$;
2. whether the node spanning $c$ has an ancestor NP node; if so, whether this NP node is a descendant of $c$'s lowest ancestor IP node;
3. whether the node spanning $c$ has an ancestor VP node; if so, whether this VP node is a descendant of $c$'s lowest ancestor IP node;
4. whether $vp$ has an ancestor NP node, where $vp$ is the VP node spanning the VP that follows $p$;
5. whether $vp$ has an ancestor VP node;

[7] We employ a simple heuristic to identify the speaker of NPs occurring in direct speech: we assume that the speaker is the subject of the speech's reporting verb. So for example, we identify *Jack* as the speaker of *This book* in the sentence *"This book is good," Jack said.*

|  | Training | Test |
|---|---|---|
| Documents | 1,391 | 172 |
| Sentences | 36,487 | 6,083 |
| Words | 756,063 | 110,034 |
| Overt Subject Pronouns | 13,418 | – |
| AZPs | – | 1,713 |

Table 7: Statistics on the training and test sets.

6. whether $p$ is the first word of a sentence; if not, whether $p$ is the first word of an IP clause;
7. whether $c$ is a subject whose governing verb is lexically identical to the verb governing $p$;
8. whether $c$ is the closest candidate antecedent with subject grammatical role and is semantically compatible with $p$'s governing verb; if not, whether $c$ is the first semantically compatible candidate antecedent[8].

Our approach to determine semantic compatibility (in feature 8) resembles Kehler et al.'s (2004) and Yang et al.'s (2005) methods for computing selectional preferences. Specifically, for each verb and each noun that serves as a subject in Chinese Gigaword, we compute their mutual information (MI). Now, given a pronoun $p$ and a candidate antecedent $c$ in the training/test corpus, we retrieve the MI value of $c$ and $p$'s governing verb. We then consider them semantically compatible if and only if their MI value is greater than zero.

## 7 Evaluation

### 7.1 Experimental Setup

**Datasets.** We employ the Chinese portion of the OntoNotes 5.0 corpus that was used in the official CoNLL-2012 shared task (Pradhan et al., 2012). In the CoNLL-2012 data, the training set and development set contain ZP coreference annotations, but the test set does not. Therefore, we train our models on the training set and perform evaluation on the development set. Statistics on the datasets are shown in Table 7. The documents in these datasets come from six sources, namely Broadcast News (BN), Newswire (NW), Broadcast Conversation (BC), Telephone Conversation (TC), Web Blog (WB) and Magazine (MZ).

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

**Evaluation settings.** Following Chen and Ng (2013), we evaluate our model in three settings. In Setting 1, we assume the availability of gold syntactic parse trees and gold AZPs. In Setting 2, we employ gold syntactic parse trees and system (i.e., automatically identified) AZPs. Finally, in Setting 3, we employ system syntactic parse trees and system AZPs. The gold and system syntactic parse trees, as well as the gold AZPs, are obtained from the CoNLL-2012 shared task dataset, while the system AZPs are identified by the rule-based approach described in the Appendix.[9] Since our AZP identification approach does not rely on any labeled data, we are effectively evaluating an end-to-end unsupervised AZP resolver in Setting 3.

### 7.2 Results

**Baseline systems.** We employ seven resolvers as baseline systems. To gauge the difficulty of the task, we employ four simple rule-based resolvers, which resolve an AZP $z$ to (1) the candidate antecedent closest to $z$ (Baseline 1); (2) the subject NP closest to $z$ (Baseline 2); (3) the closest candidate antecedent that is semantically compatible with $z$ (Baseline 3); and (4) the first candidate antecedent that is semantically compatible with $z$, where the candidate antecedents are visited according to the order described in Footnote 8 (Baseline 4). These four baselines allow us to study the role of (1) recency, (2) salience, (3) recency combined with semantic compatibility, and (4) salience combined with semantic compatibility in AZP resolution respectively. The remaining three baselines are state-of-the-art supervised AZP resolvers, which include our own resolver (Chen and Ng, 2013) as well as our re-implementations of Zhao and Ng's (2007) resolver and Kong and Zhou's (2010) resolver.

The test set results of these seven baseline resolvers when evaluated under the three aforementioned evaluation settings are shown in Table 8. The system AZPs employed by the rule-based resolvers are obtained using our rule-based

[8] We sort the candidate antecedents of $p$ as follows. We first consider the subject candidate antecedents in the same sentence as $p$ from right to left, then the other candidate antecedents in the same sentence from right to left. Next, we consider the candidate antecedents in the previous sentence, also preferring candidates that are subjects, but in left-to-right order. Finally, we consider the candidate antecedents two sentences back, following the subject-first, left-to-right order.

[9] One may wonder why we do not train a supervised system for identifying AZPs and instead experiment with a rule-based AZP identification system. The reason is that employing labeled data defeats the whole purpose of having an unsupervised AZP resolution model: if annotated data is available for training an AZP identification system, the same data can be used to train an AZP resolution system.

| | Setting 1: Gold Parses, Gold AZPs | | | Setting 2: Gold Parses, System AZPs | | | Setting 3: System Parses, System AZPs | | |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | R | P | F | R | P | F | R | P | F |
| Selecting closest candidate antecedent | 25.0 | 25.2 | 25.1 | 18.3 | 10.8 | 13.6 | 10.3 | 6.7 | 8.1 |
| Selecting closest subject | 42.0 | 43.6 | 42.8 | 31.8 | 19.2 | 23.9 | 18.0 | 11.9 | 14.4 |
| Selecting closest semantically compatible candidate antecedent | 28.5 | 28.8 | 28.7 | 20.5 | 12.2 | 15.3 | 11.7 | 7.6 | 9.2 |
| Selecting first semantically compatible candidate antecedent | 45.2 | 45.7 | 45.5 | 33.6 | 20.0 | 25.1 | 18.9 | 12.3 | 14.9 |
| Zhao and Ng (2007) | 41.5 | 41.5 | 41.5 | 22.4 | 24.4 | 23.3 | 12.7 | 14.2 | 13.4 |
| Kong and Zhou (2010) | 44.9 | 44.9 | 44.9 | 33.0 | 19.3 | 24.4 | 18.7 | 11.9 | 14.5 |
| Chen and Ng (2013) | 47.7 | 47.7 | 47.7 | 25.3 | 27.6 | 26.4 | 14.9 | 16.7 | 15.7 |

Table 8: AZP resolution results of the baseline systems on the test set.

| | Setting 1: Gold Parses, Gold AZPs | | | | | | Setting 2: Gold Parses, System AZPs | | | | | | Setting 3: System Parses, System AZPs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Best Baseline | | | Our Model | | | Best Baseline | | | Our Model | | | Best Baseline | | | Our Model | | |
| Source | R | P | F | R | P | F | R | P | F | R | P | F | R | P | F | R | P | F |
| Overall | 47.7 | 47.7 | 47.7 | 47.5 | 47.9 | 47.7 | 25.3 | 27.6 | 26.4 | 35.4 | 21.0 | 26.4 | 14.9 | 16.7 | 15.7 | 19.9 | 12.9 | 15.7 |
| NW | 38.1 | 38.1 | 38.1 | 41.7 | 41.7 | 41.7 | 15.5 | 21.7 | 18.1 | 29.8 | 24.8 | 27.0 | 6.0 | 12.2 | 8.0 | 11.9 | 13.0 | 12.4 |
| MZ | 34.6 | 34.6 | 34.6 | 34.0 | 34.2 | 34.1 | 18.5 | 19.6 | 19.0 | 24.1 | 14.5 | 18.1 | 6.2 | 9.4 | 7.5 | 6.2 | 5.2 | 5.7 |
| WB | 46.1 | 46.1 | 46.1 | 47.9 | 47.9 | 47.9 | 21.8 | 22.0 | 21.8 | 37.3 | 18.7 | 24.9 | 8.5 | 11.4 | 9.7 | 19.0 | 11.3 | 14.2 |
| BN | 47.2 | 47.2 | 47.2 | 52.8 | 52.8 | 52.8 | 21.8 | 33.2 | 26.3 | 31.5 | 28.1 | 29.7 | 14.6 | 26.3 | 18.8 | 18.2 | 19.5 | 18.8 |
| BC | 52.7 | 52.7 | 52.7 | 49.8 | 50.3 | 50.0 | 23.3 | 30.7 | 26.5 | 38.0 | 21.0 | 27.0 | 12.7 | 16.2 | 14.3 | 20.6 | 12.4 | 15.5 |
| TC | 51.2 | 51.2 | 51.2 | 45.2 | 46.7 | 46.0 | 43.1 | 28.2 | 34.1 | 42.4 | 20.3 | 27.4 | 33.2 | 17.1 | 22.5 | 32.2 | 13.3 | 18.8 |

Table 9: AZP resolution results of the best baseline and our unsupervised model on the test set.

AZP identification system. On the other hand, since our supervised resolvers are meant to be re-implementations of existing resolvers, we follow previous work and let them employ a supervised AZP identification system. In particular, we employ the one described in Chen and Ng (2013).

Several observations can be made about these results. First, among the rule-based resolvers, Baseline 4 achieves the best performance, outperforming Baselines 1, 2, and 3 by 12.9%, 1.5%, and 10.8% in F-score respectively when averaged over the three evaluation settings. From their relative performance, which remains the same in the three settings, we can conclude that as far as AZP resolution is concerned, (1) salience plays a greater role than recency; and (2) semantic compatibility is useful. Second, among the supervised baselines, our supervised resolver (Chen and Ng, 2013) achieves the best performance, outperforming Zhao and Ng's resolver and Kong and Zhou's resolver by 3.9% and 2.0% in F-score respectively when averaged over the three evaluation settings. Finally, comparing the rule-based resolvers and the learning-based resolvers, we can see that the best rule-based baseline (Baseline 4) performs even better than Zhao and Ng's resolver and Kong and Zhou's resolver.

In the rest of this subsection, we will compare our unsupervised model against the best baseline, Chen and Ng's (2013) supervised resolver.

**Our model.** Results of the best baseline and our model on the entire test set and each of the six sources are shown in Table 9. As we can see, our model achieves the same overall F-score as the best baseline under all three settings, despite the fact that it is unsupervised. In fact, our model even outperforms the best baseline on NW, WB and BN in Setting 1, NW, WB, BN and BC in Setting 2, and NW, WB and BC in Setting 3.

It is worth mentioning that while the two resolvers achieved the same overall performance, their outputs differ a lot from each other. Specifically, the two models only agree on the antecedents of 55% of the AZPs in Setting 1.[10]

### 7.3 Ablation Experiments

**Impact of $P(p_a|c_a, l{=}1)$ and $P(l{=}1|k_c)$.** Recall that our model is composed of five probability terms, $P(p_a|c_a, l{=}1)$ for each of the four grammatical attributes and $P(l{=}1|k_c)$, the context probability. To investigate the contribution of context and each attribute to overall performance, we conduct ablation experiments. Specifically, in each ablation experiment, we remove exactly one probability term from the model and retrain it.

[10] Note that it is difficult to directly compare the outputs produced under Settings 2 and 3: the AZPs identified by the best baseline are quite different from those identified by our rule-based system, as can be inferred from the AZP identification results in Table 12.

| | Setting 1 | | | Setting 3 | | |
|---|---|---|---|---|---|---|
| System | R | P | F | R | P | F |
| Full model | 47.5 | 47.9 | 47.7 | 19.9 | 12.9 | 15.7 |
| − NUMBER | 47.5 | 47.9 | 47.7 | 19.7 | 12.8 | 15.5 |
| − GENDER | 44.5 | 45.0 | 44.7 | 19.2 | 12.5 | 15.1 |
| − PERSON | 45.2 | 45.6 | 45.4 | 19.1 | 12.4 | 15.1 |
| − ANIMACY | 45.1 | 45.5 | 45.3 | 19.1 | 12.4 | 15.1 |
| − Context Features | 32.9 | 33.1 | 33.0 | 15.2 | 9.8 | 11.9 |

Table 10: Probability term ablation results.

Ablation results under Settings 1 and 3 are shown in Table 10. As we can see, under Setting 1, after NUMBER is ablated, performance does not drop. We attribute this to the fact that almost all candidate antecedents are *singular*. On the other hand, when we ablate any of the remaining three attributes, performance drops significantly by 2.3−3.0% in overall F-score.[11] Similar trends can be observed with respect to Setting 3: after NUMBER is ablated, performance only decreases by 0.2%, while ablating any of the other three attributes results in a drop of 0.6%.

Results after ablating context are shown in the last row of Table 10. As we can see, the F-score drops significantly by 14.7% and 3.8% under Settings 1 and 3 respectively. These results illustrate the importance of context features in our model.

**Context feature ablation.** Recall that we employed eight context features to encode the relationship between a pronoun and a candidate antecedent. To determine the relative contribution of these eight features to overall performance, we conduct ablation experiments under Settings 1 and 3. In these ablation experiments, all four grammatical attributes are retained in the model.

Ablation results are shown in rows 2−9 of Table 11. To facilitate comparison, the F-score of the model in which all eight context features are used is shown in row 1. As we can see, feature 8 (the rule-based feature) is the most useful feature: its removal causes the F-scores of our resolver to drop significantly by 6.4% under Setting 1 and 1.5% under Setting 3.

### 7.4 Error Analysis

To gain additional insights into our full model, we examine its major sources of error below. To focus on errors attributable to AZP resolution, we analyze our full model under Setting 1.

Specifically, we randomly select 100 AZPs that our model incorrectly resolves under Setting 1. We found that 17 errors are attributable to discourse disfluency, lack of background knowledge and subject detection, while the remaining 83 errors can be divided into three types:

| | Setting 1 | | | Setting 3 | | |
|---|---|---|---|---|---|---|
| System | R | P | F | R | P | F |
| Full model | 47.5 | 47.9 | 47.7 | 19.9 | 12.9 | 15.7 |
| − Feature 1 | 46.1 | 46.5 | 46.3 | 19.4 | 12.6 | 15.3 |
| − Feature 2 | 46.5 | 46.9 | 46.7 | 19.4 | 12.6 | 15.3 |
| − Feature 3 | 45.3 | 45.7 | 45.5 | 19.1 | 12.4 | 15.1 |
| − Feature 4 | 47.4 | 47.8 | 47.6 | 20.1 | 13.0 | 15.8 |
| − Feature 5 | 47.4 | 47.8 | 47.6 | 19.7 | 12.8 | 15.5 |
| − Feature 6 | 47.1 | 47.5 | 47.3 | 19.6 | 12.7 | 15.4 |
| − Feature 7 | 47.1 | 47.5 | 47.3 | 20.1 | 13.0 | 15.8 |
| − Feature 8 | 41.2 | 41.6 | 41.4 | 18.0 | 11.8 | 14.2 |

Table 11: Context feature ablation results.

**Failure to recognize the topics of a document.** Our model incorrectly resolves 32 AZPs that are coreferent with NPs corresponding to the topics of the associated documents. Consider the following example:

[八里乡] 位于台北盆地西北端。行政区隶属于台北县，*pro* 为台北县廿九个乡镇市之一。([Bali Town] is located in the Northwest of Taipei Basin. Its administrative area is affiliated with Taipei County, *pro* is one of Taipei County's 29 towns and cities.)[12]

The model incorrectly resolves the AZP *pro* to 行政区 (Its administrative area). The reason is that the correct antecedent, 八里乡 (Bali Town), is far from *pro*: there are five candidate antecedents between *pro* and 八里乡 (Bali Town). Note, however, that it is easy for a human to resolve *pro* to 八里乡 (Bali Town) because the whole passage is discussing 八里乡 (Bali Town). Hence, to correctly handle such cases, one may construct a topic model over the passage and assign each candidate antecedent a prior probability so that the resulting system favors the selection of candidates representing the topics as antecedents.

**Errors in computing semantic compatibility.** This type of error contributes to 28 of the incorrectly resolved AZPs. When computing semantic compatibility in our model, we only consider the mutual information between a candidate antecedent and the pronoun's governing verb, but in some cases, additional context needs to be taken into account. Consider the following example:

[11] All significance tests are paired $t$-tests, with $p < 0.05$.

[12] The pronoun *Its* in the phrase *Its administrative area* is inserted into the English translation for the sake of grammaticality and correct understanding of the sentence. The corresponding Chinese phrase does not contain any pronoun.

[一支海军陆战队] 杀死了约 [24 名手无寸铁的伊拉克人]，*pro* 包括妇女和六名儿童。
([Marines] killed about [24 unarmed Iraqis], *pro* include women and six children.)

There are two candidate antecedents in this example, 一支海军陆战队 (Marines) and 24 名手无寸铁的伊拉克人 (24 unarmed Iraqis), which we denote as $c_1$ and $c_2$ respectively. The correct antecedent of *pro* is $c_2$ , while our model wrongly resolves *pro* to $c_1$. Note that both $c_1$ and $c_2$ are compatible with the AZP's governing verb 包括 (include). However, if the object of the governing verb, i.e., 妇女和六名儿童 (women and six children), were also considered, the model could determine that $c_1$ is not compatible with the object while $c_2$ is, and then correctly resolve *pro* to $c_2$.

**Failure to recognize and exploit semantically similar sentences.** This type of error contributes to 23 wrongly resolved AZPs. Recall that an AZP is omitted for brevity, so the sentence it appears in often expresses similar meaning to an earlier sentence. However, our model fails to handle such cases. Consider the following example:

[指挥部和突进的部队] 之间也会失去联络。..... *pro* 就联络不上了。
([The command and the onrush of troops] lost connection with each other. ... *pro* cannot connect with each other.)

The above example shows two sentences that are separated by some other sentences. The AZP under consideration is in the last sentence, while the first sentence contains the correct antecedent 指挥部和突进的部队 (the command and the onrush troops), denoted as $c_1$. Our model fails to resolve *pro* to $c_1$, because there are many competing candidate antecedents between $c_1$ and AZP. However, if our model were aware of the similarity between the constructions appearing after $c_1$ and *pro*, i.e., 之间也会失去联络 (lost connection with each other) and 就联络不上了 (cannot connect with each other), then it might be able to correctly resolve the AZP.

## 8 Conclusion

We proposed an unsupervised model for Chinese zero pronoun resolution, investigating the novel hypothesis that an unsupervised probabilistic resolver trained on overt pronouns can be applied to resolve ZPs. To our knowledge, this is the first unsupervised probabilistic model for this task. Experiments on the OntoNotes 5.0 corpus showed that our unsupervised model rivaled its state-of-the-art supervised counterparts in performance.

## Acknowledgments

We thank the three anonymous reviewers for their detailed comments. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142.

## References

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 33--40.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 148--156.

Chen Chen and Vincent Ng. 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 Chen and Vincent Ng. 2014. SinoCoreferencer: An end-to-end Chinese event coreference resolver. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4532--4538.

Colin Cherry and Shane Bergsma. 2005. An expectation maximization approach to pronoun resolution. In *Proceedings of the Ninth Conference on Natural Language Learning*, pages 88--95.

Susan Converse. 2006. *Pronominal Anaphora Resolution in Chinese*. Ph.D. thesis, University of Pennsylvania.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1--38.

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.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203--226.

Na-Rae Han. 2006. *Korean Zero Pronouns: Analysis and Resolution*. Ph.D. thesis, University of Pennsylvania.

Jerry Hobbs. 1978. Resolving pronoun references. *Lingua*, 44:311--338.

Ryu Iida and Massimo Poesio. 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*, pages 804--813.

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.

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

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 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.

Hideki Isozaki and Tsutomu Hirao. 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.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 289--296.

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.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2009. Chinese Gigaword fourth edition. Linguistic Data Consortium. Philadelphia, PA.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Shared Task*, pages 1--40.

Xian Qian, Qi Zhang, Xuangjing Huang, and Lide Wu. 2010. 2d trie for fast parsing. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 904--912.

Ryohei Sasano and Sadao Kurohashi. 2011. A 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.

Kazuhiro Seki, Atsushi Fujii, and Tetsuya Ishikawa. 2002. A probabilistic method for analyzing Japanese anaphora integrating zero pronoun detection and resolution. In *Proceedings of the 19th International Conference on Computational Linguistics*.

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.

Ching-Long Yeh and Yi-Chun Chen. 2007. Zero anaphora resolution in Chinese with shallow parsing. *Journal of Chinese Language and Computing*, 17(1):41--56.

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.

## Appendix: Automatic AZP Identification

Our automatic AZP identification system employs an ordered set of rules. The first rule is a positive rule that aims to extract as many *candidate* AZPs as possible. It is followed by seven negative rules that aim to improve precision by filtering out erroneous candidate AZPs. Below we first describe the rules and then evaluate this rule-based system.

**Rule 1.** Add candidate AZP $z$ if it occurs before the leftmost word spanned by a VP node $vp$.

**Rule 2.** Remove $z$ if its associated $vp$ is in a coordinate structure or modified by an adverbial node.

**Rule 3.** Remove $z$ if the parent of its associated $vp$ node is not an IP node.

**Rule 4.** Remove $z$ if its associated $vp$ has a NP or QP node as an ancestor.

**Rule 5.** Remove $z$ if one of the left sibling nodes of $vp$ is NP, QP, IP or ICP.

**Rule 6.** Remove $z$ if (1) $z$ does not begin a sentence, (2) the highest node whose spanning word sequence ends with the left non-comma neighbor word of $z$ is either NP, QP or IP, and (3) the parent of this node is VP.

| Systems | Gold Parses | | | System Parses | | |
|---|---|---|---|---|---|---|
| | R | P | F | R | P | F |
| Rule-based | 72.4 | 42.3 | 53.4 | 42.3 | 26.8 | 32.8 |
| Supervised | 50.6 | 55.1 | 52.8 | 30.8 | 34.4 | 32.5 |

Table 12: AZP identification results on the test set.

**Rule 7.** Remove $z$ if $vp$'s lowest IP ancestor has (1) a VP node as its parent and (2) a VV node as its left sibling.

**Rule 8.** Remove $z$ if it begins a document.

To gauge the performance of our rule-based AZP identification system, we compare it with our supervised AZP identification system (Chen and Ng, 2013). Results of the two systems on our test set are shown in Table 12. As we can see, the F-scores achieved by the rule-based system is comparable to those of the supervised system.