Visualizing Zeros: A Japanese Learning Aid

Mitsuko Yamura-Takei, Makoto Yoshie and Teruaki Aizawa
Graduate School/Faculty of Information Sciences, Hiroshima City University
{yamuram, yoshie, aizawa}@nlp.its.hiroshima-cu.ac.jp

1 Introduction

Zero anaphora or zero pronouns (henceforth zeros) are referential noun phrases that are not overtly expressed in Japanese discourse. The use of zeros is common in Japanese and this has been an intriguing research topic in various disciplines.

Zero anaphora resolution has attracted much attention among NLP researchers and a variety of methods have been proposed to resolve anaphoric relations between zeros and their antecedents. Whatever the approach may be, it involves two steps: a zero identification phase followed by a zero interpretation phase. The first step is a prerequisite, though it is not necessary for an overt-pronoun language like English. However, the primary focus has always been on the second step, and the identification process has been largely neglected in past research. Zeros are often manually detected as a preparatory process for zero interpretation. No full description of the implementation of such a program has been provided, nor has its performance been reported in any study to date, to our best knowledge. Yet automatic detection of "invisible" zeros is not an easy and straightforward task, but involves complex processes that deserve focused, intensive research efforts.

From a pedagogical perspective, on the other hand, zeros often pose a major challenge for Japanese as a Second Language (JSL) learners in their accurate comprehension and natural-sounding production of Japanese sentences with zeros. Some learners fail to understand a passage correctly because of the difficulty identifying zeros and/or their antecedents. Other learners produce unnatural-sounding Japanese due to overuse or underuse of zeros. Native speakers of Japanese usually learn zeros without any conscious process. Second language learners, however, might benefit more from explicit instruction or awareness-raising than from just waiting for natural acquisition, as some recent second language acquisition principles suggest (see Norris and Ortega, 2000).

In light of all this, we have developed a Japanese learning aid, Zero Detector. Zero Detector (hereafter ZD) is an automatic zero identifying tool, which takes Japanese written narratives as input and provides zero-specified texts as output. This aims to draw learners' attention to zeros, by making these invisible elements visible. ZD employs a rule-based approach, with theoretically sound heuristics. We have integrated two existing natural language analysis tools and an electronic dictionary, none of which were intended for a language learning aid, into the program, attempting to make the best possible use of their capabilities for our purpose.

In the next section, we outline the theoretical assumptions from which our heuristics are drawn. We go on to describe the zero detecting processes that ZD goes through. Next, we present the results of the system evaluation. After we briefly discuss its application to JSL learning, we conclude with some suggestions for future work.

2 Theoretical Assumptions

As we mentioned in the previous section, ZD is a heuristic, rule-based program. In this section, we present some linguistic assumptions upon which our heuristics are based.

2.1 Predicate-Argument Structure

Japanese is a head-final language. A sentence or a clause is headed by a predicate, which takes a set of arguments and adjuncts. Predicates in Japanese include verbs, adjectives, nominal adjectives and copula, usually consisting of a core predicate and some auxiliary elements. Arguments are classified into three types: Topic Phrase (TP), headed by a topic marker '*wa*', Focus Phrase (FP), headed by focus particles '*mo, koso, dake*, etc.', and Kase Phrase (KP), headed by case particles '*ga, wo, ni, e, to, yori, de, kara*, and *made*.' We regard adjuncts as non-particle-headed phrases.

2.2 Obligatory Arguments and Zeros

We define zeros as unexpressed obligatory arguments of a core predicate. What is "obligatory" is the next question to arise. Obligatoriness is a controversial issue, and there is no set agreement among linguists on its definition. Somers (1984) proposed a six-level scale of valency binding that reflects the degree of

closeness of an element to the predicate. The levels are (i) integral complements, (ii) obligatory complements, (iii) optional complements, (iv) middles, (v) adjuncts and (vi) extraperipherals. Ishiwata (1999) suggests that in Japanese group (i) is often treated as part of idioms and is not omissible, and Japanese nominative *–ga* and accusative *–o* fall into the categorty (ii), while dative *–ni* belongs to (iii). In light of this, we assume that obligatory arguments that can be zero-pronominalized are phrases headed by nominative-case particles *–ga* and *–ni* and accusative *–o*, *-ni*, and *–ga*, excluding dative *–ni*.

3 Zero Detecting Processes

ZD goes through the following five main steps to achieve its goal.

3.1 Morphological Analysis and Clause Splitting

Input text is first morphologically analysed by ChaSen 2.2.8 (NAIST, Matsumoto, Y. *et al.*, 2001). The text is then divided into clauses, each consisting of one and only one predicate and its arguments. Some predicates are simplex, while others are complex, consisting of more than one core predicate (i.e., '*jiritsu*' predicates in ChaSen analysis). Some complex predicates (e.g., *tabeta-koto-ga-aru*) are predefined as simplex to avoid excessive clause splitting. The clauses are then classified into three categories: independent (main), dependent (coordinated /subordinated) and embedded clauses. A clause serves as the basic unit for a zero detecting operation. In this study, embedded clauses (relative/nominal/quoted) are excluded from this operation and are left within their superordinate clauses.

3.2 Constructing a Clause Structure Frame

Once text is split into clauses, each clause is analysed for its dependency structure by CaboCha 0.21 (NAIST, Kudo, K., 2001) and then converted into its clause structure frame. An example of this frame is given in Figure 1.

3.3 Valency Checking

A core predicate is checked against the Goi-Takei Valency Dictionary (henceforth GTVD; Ikehara *et al.*, 1997) to search for its syntactic valency pattern. GTVD is a semantic valency dictionary, originally designed for transfer-based Japanese-to-English machine translation, so it includes as many valency pattern entries for each predicate as are necessary for effective transfer. The entries are ordered according to expected frequency of occurrence. So we decided to take the very naïve approach of selecting the first-ranking entry from the listing for each core predicate.

Input: 数日後、奥さんはこまかいことを調べさせました。
Paragraph#: 2
Sentence#: 4
Clause#: 5
Clause Type: Independent

[Predicate]: 調べさせました。
Core: 調べる 動詞-自立 未然形
Auxiliary: させる 動詞-接尾 連用形
ます 助動詞 連用形
た 助動詞 基本形
。 記号—句点
Voice: causative
Empathy:
Conjunction:

[Argument]:
Topic Phrase: 奥さんは
Topic-Case: N1が
Focus Phrase: <none>
Focus-Case: <none>
Kase Phrase: こまかいことを
Pre-copula: <none>
[Adjunct]: 数日後

Figure 1: A clause structure frame

The next step is to apply our own definition of 'obligatoriness' described in 2.2, to refine a selected valency pattern. If non-*ga*, *wo*, or *ni* cases are within the first three case slots, they are excluded. If a *ni*-case still remains in the third case slot, it is also deleted. These operations leave us only two valency patterns: (i) N1-*ga* N2-*wo*, and (ii) N1-*ga* N2-*ni*, in most cases.

Lastly, as valency-changing derivations, passives and causatives are considered. In the particular example in Figure 2, N3-*ni* is added to the valency because the voice slot is marked as causative in Figure 1.

Valency Selected:	N1が	N2を	
Valency Obligatory:	N1が	N2を	
Valency Changed:	N1が	N2を	N3に

Figure 2: Valency checking for example in figure 1

3.4 Identifying Zeros

Now that the valency pattern for the given predicate is assigned, it is checked against overt arguments listed in the frame. The matched valent (N2 in Figure 3) is removed from the zero candidates.

Valency Changed: ~~N1が~~ ~~N2を~~ N3に Zero: N3に

Figure 3: Valent matching

Case-less elements, such as TP and FP, should first restore their cases. This is done by assigning the first remaining valent to TP and/or to FP. Again, this is based on the linguistic fact that subjects are far more likely to be topicalized or focused than objects. In the example, TP is assigned *ga* case. The assigned case slot (N1 in Figure 3) is also deleted.

Finally, the remaining valent, if any, is assumed to be a zero (N3 in Figure 3).

3.5 Specifying Zeros

Once zeros are identified, it is necessary to decide where to insert identified zeros in the original text. This is done by keeping canonical ordering, as listed in a valency pattern. An example predicate-(obligatory) argument structure from Figure 1, with an identified zero, is presented in Figure 4.

*奥さん は (が) *こまかいこと を *[に] *調べさせました。

Figure 4: Predicate-argument structure with zero(s)

Finally, ZD outputs the original series of clauses with zeros inserted in the most plausible positions, along with adjuncts, as in Figure 5.

数日後、奥さんは こまかいことを [に] 調べさせました。

Figure 5: Zero-specified text

3.6 Architecture Overview

In sum, ZD produces four different types of output: (1) split clauses, (2) clause structure frames, (3) predicate-argument structures with zeros, and (4) zero-inserted text. Output (1) can be manually corrected, when necessary, before it is entered into the zero detecting module. The flow of the system is illustrated in Figure 6.

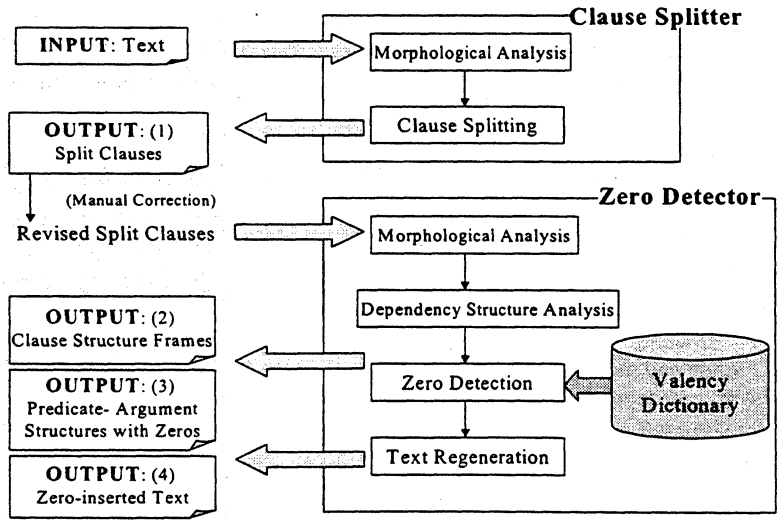

Figure 6: Flow diagram of zero detecting process

4 Evaluation

The purpose of the evaluation was to assess the validity of ZD output for its practical use in a language learning setting.

4.1 Methodology

The test corpus consisted of two lower-intermediate reading materials and one student writing sample, all of which were written narratives. Five subjects (native speakers of Japanese and trained natural language researchers) served as human zero detectors. They

were asked to intuitively identify missing arguments in each clause. We used average human performance as a baseline against which to evaluate ZD output. Here, zeros detected by three or more, out of five, subjects were regarded as average human performance.

4.2 Result and Discussion

As Table 1 shows, ZD achieved a 73% per-clause matching rate with human output. That number represents the ratio of the exact match between two outputs over the total number of clauses.

Table 1: Per-clause matching rates

	# of clauses	# of matched
Reading (1)	30	22 (73%)
Reading (2)	25	18 (72%)
Writing	23	17 (74%)
Total	78	57 (73%)

Closer examination of each case is given in Table 2.

Table 2: Per-case element matching rates

		が ga		を wo		に ni	
		Human	ZD	Human	ZD	Human	ZD
Matched	Detected	35	32	5	4	5	2
	Not Detected	43	39	73	68	73	63
	Total	78	**71(91%)**	78	**72(92%)**	78	**65(83%)**
Not Matched	Under-detected		3		1		3
	Over-detected		4		5		10
	Total		**7(9%)**		**6(8%)**		**13(15%)**

The level 'matched' includes both cases where ZD and human detect a zero and cases where neither detects it. The accuracy is high enough for the ZD output to be put into practical use as a learning aid, without an excessive load on teachers for correcting output errors.

We analysed 'unmatched' cases to improve future performance. There are a few cases of both underproduction and overproduction of zeros. Many of them are caused by our naïve valency selection algorithm. Another cause is our case-restoring heuristics. They sometimes do not function properly when accusatives or adjuncts are topicalized (or focused).

5 Application to JSL Learning

ZD has been designed as a language learning aid although it can also serve as a pre-processing module for anaphora resolution and other NLP applications. ZD provides JSL teachers and learners with predicate-argument structures and text with "visible" zeros. These output texts were experimentally used in a university-level intermediate JSL classroom through digital presentation. The students showed a positive reaction to this analytic instruction. They described this approach as "effective" and "clear" for understanding zeros.

6 Future Work

ZD is currently a purely syntactic-based tool that utilizes only surface-level heuristics, excluding any semantic information. As our error analysis in Section 4 obviously indicates, more accuracy can be achieved in a semantically-enhanced version, which in fact is our next project goal. Valency-pattern-selecting (from GTVD) and case-restoring (from TP and FP) algorithms are two major areas to which semantic information can greatly contribute.

References

Ikehara, S., M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura and Y. Hayashi (1997). *Goi-Takei – A Japanese Lexicon*, 5 volumes, Iwanami Shoten, Tokyo.

NAIST, Kudo, K. (2001). *CaboCha* 0.21. http://cl.aist-nara.ac.jp/~taku-ku/software/cabocha/

NAIST, Matsumoto, Y. *et al.* (2001). *ChaSen* 2.2.8. http://chasen.aist-nara.ac.jp/

Ishiwata, T. (1999). *Gendai GengoRiron to Kaku*, Hituzi Shobo, Tokyo.

Norris, J.M. and L. Ortega (2000). Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis. *Language Learning* 50 (3), pp.417-528.

Somers, H.L. (1984). On the validity of the complement-adjunct distinction in valency grammar. *Linguistics* 22, pp. 507-53.