Disambiguating between ‘wa’ and ‘ga’ in Japanese

Yoshihiro Komori
500 College Avenue
ykomori1@swarthmore.edu

Abstract

This paper attempts to distinguish when to use ‘wa’ and ‘ga’ in Japanese. The problem is treated as one of word sense disambiguation, regarding both ‘wa’ and ‘ga’ as a prototype particle that indicates the subject of the sentence. Various statistical and linguistic techniques are employed to disambiguate the sense, such as ngram and syntactic analysis. The program scored 100% recall rate and 83.8% using the syntactic model.

1 Introduction

The distinction between ‘wa’ and ‘ga’ in Japanese has been notoriously hard for both Japanese linguists and those who attempt to learn Japanese as a foreign language. Both ‘wa’ and ‘ga’ are particles to indicate the subject but with slightly different connotations. For example, the English sentence

I am Yoshi.

can be translated to Japanese as

watashi	***wa***	*yoshi*	*desu.*
I	(null)	Yoshi	am.

or

watashi	***ga***	*yoshi*	*desu.*
I	(null)	Yoshi	am.

Whether we should use ‘wa’ or ‘ga’ cannot be determined locally in many cases, such as in this example. Those two Japanese sentences are syntactically valid and commonly used. To determine which particle to use, we need first determine the semantics of the sentence from its context.

There are several areas where having a machine that distinguishes ‘wa’ and ‘ga’ can be helpful. Those include translation to Japanese from various languages (given a sentence in a foreign language, should we use ‘wa’ or ‘ga’), Japanese sentence generation (given what we want to say, should we use ‘wa’ or ‘ga’), and Japanese linguistic theory (what are the conditions that require the use of ‘wa’ or ‘ga’?).

2 Linguistic Theory

Karita Shuji, a Japanese linguist, summarized the works of recent linguists on the usage of ‘wa’ and ‘ga’. According to Karita, the usage of ‘wa’ and ‘ga’ can be categorized in the following manner:

2.1 Substitution by ’no’

In some cases ‘wa’ can be substituted by ’no’ without changing the meaning of the sentence, but ‘ga’ can never be. This is due to the fact that the noun preceded by ‘wa’ does not have to be the actor in the sentence.

2.2 Novelty of the subject

One sense of ‘wa’ and of ‘ga’ is dependent on whether the subject is novel to the listener. While ‘wa’ is used when the subject is novel, ‘ga’ is used when the subject is known, previously mentioned or implied. Even when the subject is a novel topic in the discourse, ‘ga’ might be used if the information is implied by information outside of the discourse. An instance of such cases is:

watashi	***ga***	*senjitsu*	*email*	*shita*
I	(null)	the other day	email	did
gakusei	*desu.*			
student	am.			

(I am the student that emailed you the other day.)

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 30–34
Computer Science Department, Swarthmore College

The fact of the subject's emailing the listener makes the subject a familiar topic even though it may be the first time it is introduced in the discourse.

2.3 Description vs Judgment

Karita argues that in some cases 'ga' is used for describing a phenomenon, while 'wa' is used for expressing a judgment. For example,

ame	***ga***	*futte*	*iru*
rain	(null)	raining	is

(Rain is falling.) (*description*)

are	***ga***	*ume*	*da.*
that	(null)	plum	is.

(That is a plum.) (*judgment*)

In the first case we use 'ga', and in the second case 'wa'. The difference, however, is slight and is hard even for a native speaker to distinguish.

2.4 Sentence Structure

We use 'wa' if the subject specified is the subject of the entire sentence, and we use 'ga' if the subject is only the subject of a phrase in the sentence. so, for example:

tori	***ga***	*tobu*	*toki*	*ni*	*wa*
bird	(null)	fly	when	(null)	(null)
kuuki	*ga*	*ugoku.*			
air	(null)	move.			

(When a bird flies, the air moves.)

tori	***wa***	*tobu*	*toki*	*ni*	
bird	(null)	fly	when	(null)	
hane	*wo*	*konna*	*fuu*	*ni*	*suru.*
wing	(null)	like this	way	(null)	do.

(A bird moves its wings like this when it flies.)

The bird in the first sentence is a subject only in a phrase, where the second bird is the subject of the entire sentence. Note that being the subject of an entire sentence is not a necessary condition for using 'wa'. However, being a subject inside a phrase is a necessary condition for using 'ga'. Therefore, if 'wa' or 'ga' is to be used inside a phrase, 'ga' must be used all the time.

2.5 Contrast and Exclusion

Karita argues that we use 'wa' to indicate contrast and 'ga' to indicate exclusion. Two exampler sentences:

ame	***wa***	*futte*	*iru*	*ga*
rain	(null)	fall	(-ing)	but
yuki	***wa***	*futte*	*inai.*	
snow	(null)	fall	(not -ing)	

(Rain is falling but snow isn't)

yoshi	***ga***	*seito*	*desu*
Yoshi	(null)	student	is.

(Yoshi is the student.)

In the first sentence, 'wa' is used to express the contrast between 'rain' and 'snow,' while in the second sentence 'ga' is used to imply that Yoshi is the only student in the context.

2.6 Specimen

Two sentences:

chou	***wa***	*mushi*	*da.*
butterfly	(null)	insect	is.

(A butterfly is an insect.)

kore	ga	*kimino*	*chou*	*da.*
this	(null)	your	butterfly	is.

(This is your butterfly.)

In the first sentence 'wa' is used, and 'ga' is used for the second case. The difference between the two cases is that in the first sentence, a butterfly is a specimen of the class insect, where in the second case 'this' and 'butterfly' are in the same class.

2.7 Implication for the project

Karita's linguistic analysis on the usage of 'wa' and 'ga' has two implications for this project. First, these characterizations imply that the usage of 'wa' and 'ga' are determined by a mixture of syntactic and contextual information. Therefore, in order to capture the usage of 'wa' and 'ga', both syntactic and contextual approach must be employed. Second, from these characterization one could argue that both 'wa' and 'ga' have several different senses. This implies that in order to achieve the competence of a native speaker, the problem has to be understood as disambiguating the sense of the prototype subject indicator

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 30–34
Computer Science Department, Swarthmore College

into a dozen senses. However, such a project would require a huge tagged corpus where each instance of 'wa' and 'ga' is disambiguated from several senses. Employing humans to hand-tag such a corpus would be expensive. Thus we will attempt to disambiguate the prototype subject indicator into only two senses, 'wa' and 'ga'.

3 Related Works

The task of word sense disambiguation has been tackled by many NLP researchers, such as Yarowsky (1992, 1993 and 1995). However, the two key assumptions often made in the task of word sense disambiguation do not hold in this particular task. The assumption of 'one sense per discourse' clearly does not hold here because both 'wa' and 'ga' occur in discourses with any topic and style. The other key assumption, 'one sense per collocation,' does not hold here as well as it does in other places, since both 'wa' and 'ga' can follow any noun. On the other hand, the idea of 'one sense per collocation' can be useful if we take the collocation to be mainly syntactic and use it loosely to aid other algorithms.

4 Task Description

The input to this program consists of Japanese copra tagged with the POS. The tags are produced by a GPL engine "mecab" developed by Kudo Taku, which claims to achieve 90% accuracy. At the preprocessing stage we replace every instance of 'wa' and 'ga' that are particles to indicate the subject, determined by the POS, with a prototype particle *prt*. The output of the program is 'wa' or 'ga' for each instance of *prt*. The performance is measured by the ratio of correct determination of 'wa' and 'ga'.

The training corpus consists of three novels by a Japanese novelist Dazai Osamu, *NingenShikkaku*, *Joseito* and *Shayou*. The testing corpus consists of two other short stories by the same author, *Haha* and *Hashire Merosu*. The size of each corpus was about 130,000 words and 11,000 words respectively.

5 Algorithms

The algorithms employed in this project can be broadly divided into three parts: word based, syntactic based and context based. For word based analysis, simple ngrams are used to get as much information out of words that surround *prt*. For syntactic analysis, both ngrams with POS and sentence-level syntactic analysis are employed. Finally for context, we will test whether the word preceding *prt* is novel in the discourse.

5.1 Word Ngrams

First we used unigram on 'wa' and 'ga' on our training corpus to obtain the ratio between the occurrence of 'wa' and 'ga'. This ratio was used as the baseline in the determination of the particles. That is, if there are no further information available on the particular instance of *prt*, we will put whichever particle that has the higher ratio of occurrence.

We also use word based bigrams to get information as to whether 'wa' and 'ga' are likely to be preceded by certain words. Upon each instance of *prt*, we see what the preceding word is, and check how many times 'wa' and 'ga' have occurred in the particular context. If there is a difference in the number of occurrences, we will pick the one with the higher occurrence.

5.2 Syntactic Ngrams

Similar to the word based ngrams, we first compile POS based ngrams on training copra. Then for each instance of *prt* in the testing corpus, find the ratio of 'wa' and 'ga' in that particular POS surroundings. So far we have only considered the word preceding and the word following *prt*. A wider ngram may be employed in the future work.

5.3 Threshold for ngrams

In combining these two ngram methods, we used a threshold to increase precision and also to minimize the effect of infrequent ngrams. The algorithm used for the threshold is the following:

```
if ( countwa + 3 > 2 * ( countga + 3) )
    return wa
else if ( countga + 3 > 2 * ( countwa + 3 ) )
    return ga
else
    return (no answer)
```

(countwa and countga are the counts of the particular contexts for 'wa' and 'ga' in the corresponding ngram data.)

We first added 3 to both the count of 'wa' and 'ga' so that contexts with low counts will not produce extreme ratio. For example, while the ratio between 0 and 1 is infinity, by adding 3 to both we get a more reasonable ratio of 3/4. For either ngram method to return an answer, we required that the count of the more frequent ngrams has to be greater than twice the count of the less frequent ngrams.

5.4 Syntactic Analysis

From Karita's work we know that if the subject is the subject of a phrase but not of the sentence, then 'ga' is always to be used and not 'wa'. We will implement this model by

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 30–34
Computer Science Department, Swarthmore College

	recall	precision
wa	67.5%	86.2%
ga	60.4%	60.5%
total	65.7%	80.4%

Table 1: Performance with word based bigram analysis

	recall	precision
wa	92.2%	87.0%
ga	94.0%	63.5%
total	92.6%	81.1%

Table 2: Performance with syntactically based trigram analysis

sentence level syntactic analysis. Finding sentence structures requires a full blown syntactic analyzer, which is difficult and complex enough to dedicate a whole project. Therefore, instead of designing a thorough syntactic analyzer, we will use a simple heuristic to determine a phrase in a given sentence. The heuristic exploits the fact that a phrase in Japanese is often segmented by a comma and always contain a verb phrase. Therefore, a prototype is considered to be inside a phrase if and only if a verb phrase occurs after the prototype and before a comma.

5.5 Contextual Analysis

One sense of 'ga' indicates the fact that the subject is not a novel topic. Thus by searching through the corpus and testing to see whether the subject has been mentioned previously, we can bias the output towards 'ga'.

6 Results

We counted the occurrences of 'wa' and 'ga' to obtain the baseline for this project. In our corpus of 130,000 words, 'wa' occurred 3439 times and 'ga' occurred 2138 times. Therefore, if the program guessed 'wa' for all instances of *prt*, we can expect it to be correct 62% of the time. The baseline in this project is thus considered to be 62%.

The word based bigram model yielded results with poor recall of 65.7% but precision at 80.4% which is significantly better than the baseline. The syntactically based trigram analysis achieved slightly better precision of 81.1% and huge improvement on recall of 92.6%. Guessing was not allowed for these tests. Therefore, if the context of *prt* did not match any ngram in the data, the program did not return an answer. Thresholds are not used for these tests, either. The recall rate here is calculated as the ratio between the count of guesses for 'wa' and 'ga' and the count of the occurrences of either particle in the corpus. The precision rate is the ratio between the count of correct guesses and the count of total guesses. For example, if 'wa' occurred 10 times in the corpus, the program returned 6 guesses for 'wa', and 4 of them were correct, the recall rate would be 6/10 and the precision would be 4/6. These results are summarized in Table 1 and Table 2 respectively.

Two models gave the same answer 88.4% of the time. When the answers were in conflict, the syntactically model was correct 55.9% of the time.

	recall	precision
wa	82.6%	88.4%
ga	65.7%	76.1%
total	78.4%	85.9%

Table 3: Performance with both syntactically and word based ngram analyses

These two ngram methods combined with the threshold algorithm described above yielded results that are better in precision but worse in recall compared to the results from syntactic ngrams alone. The improvement on the precision rate on 'ga' is significant, changing from 63.5% in the syntactic ngrams approach to 76.1% in the combined methods. When two models gave different answers, the answers given by the syntactic method was always chosen. The results are summarized in Table 3.

The same algorithm but with random guesses produced results only slightly poorer in precision. Note that the precision rates for the models with and without random guesses are exactly the same. This is due to the fact that all random guesses were 'wa' since 'wa' generally occurs more frequently. The results are in Table 4.

The syntactic method based on the analysis of phrases in a sentence gave poor results. When used alone, it predicted correctly the instances of 'ga' 56.2% of the time. When used to disambiguate the cases where the word based and the syntactic based gave conflicting answers, the precision dropped to 28%.

The contextual method was a complete failure. It turned out that in almost all cases, the word preceding *prt* is introduced prior in the corpus in other contexts. In the testing corpus, only one word preceding *prt* was a novel word in the discourse. Because of this unexpected result, no further analysis was possible concerning the distinction between a novel and a familiar topic.

	recall	precision
wa	100%	85.2%
ga	100%	76.1%
total	100%	83.8%

Table 4: Performance with both syntactically and word based ngram analyses with random guesses

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 30–34
Computer Science Department, Swarthmore College

7 Discussion

The poor recall rate of word based bigram model can be attributed to the fact that the bigram data compiled from the training corpus did not contain most of the proper nouns that occurred in the testing corpus. This is an irredeemable flaw with the word based model. Because both 'wa' and 'ga' can follow any proper noun, it is inherently impossible to capture a significant portion of them. The precision rate for the case of 'wa' was surprisingly high. A closer look at the bigram data revealed that 'wa' uniquely followed two common particles, 'de' and 'ni', both of which combined with 'wa' indicate topic introduction. The precision was high due to the fact that if the preceding word were 'de' or 'ni', the prediction of 'wa' yields almost 100% accuracy.

The higher recall rate for the syntactic model was expected, since the part of speech tags are vastly more general than words. It was interesting to see that its precision rate was also higher than the word based model, which is contrary to the expectation regarding its generality. We can attribute the unexpected precision rate to the fact that this simple trigram model embodies many of the characterizations put forth by Karita. First, the rule of substitution by 'no' is reflected in the trigram model because 'no' in this sense happens only between two nouns. Second, it is seen in the trigram data that 'ga' is much more likely to be followed by a verb, which is perhaps due to the fact that 'ga' happens inside a phrase where 'noun-ga-verb' phrase is common.

The precision rate of 56.2% for the sentential syntactic analysis is statistically significant even though its absolute value is low. If we recall that the percentage of the occurrence of 'ga' among all occurrences of *prt* is only 25%, answering correctly 56.2% implies that the implementation captured at least some part of what Karita argues about 'ga' inside a phrase.

It is also worth noting that the testing corpus had an unusual distribution of 'wa' and 'ga'. Where the distribution in the much larger training corpus was about 3 to 2, the distribution in the testing corpus was 3 to 1. This unusual trend might have affected the result one way or the other. Further testing with a different corpus is required to examine the effect.

8 Conclusion

With the combination of word ngrams, syntactic ngrams and phrase analysis alone, we have achieved 83.8% precision with 100% recall. This is promising considering the fact that we did not use a syntactic analyzer outside of our heuristics. With such an aid, we can perform a complete analysis of sentential structure, which will probably boost the precision to the high 80's. With further work with a syntactic analyzer, we will perhaps be able to disambiguate all instances of 'wa' and 'ga' that have distinct syntactic contexts.

The project did not succeed in disambiguating the cases where deeper contextual analysis is required. The problem of contexts and semantics beyond pure statistics of words is a notoriously difficult one across all NLP fields. Thus we do not expect that we will be able to solve the problem without employing an entirely novel method yet undiscovered in the field of NLP. However, we do believe that using implementations similar to the current one can contribute to practical applications such as machine translations and grammar checking in Japanese. Even though by word based ngrams and syntactic analysis alone cannot capture all occurrences of 'wa' and 'ga,' they can give correct answers most of the time for most of the cases.

9 references

David Yarowsky, 1992. *Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Copra*

David Yarowsky, 1993. *One Sense Per Collocation*

David Yarowsky, 1995. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*

Karita Shuji, *'wa' to 'ga' wo meguru shomondai ni tsuite*, (http://bakkers.gr.jp/ karita/report/report$_kanyou-j.html$)

$KudoTaku(http://cl.aist-nara.ac.jp/ taku-ku/software/mecab/)$

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 30–34
Computer Science Department, Swarthmore College