Correction of Article Errors in Machine Translation Using Web-based Model

Jing PENG Kenji ARAKI
Language Media Laboratory, Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo
{pj, araki}@media.eng.hokudai.ac.jp

Abstract-**In this paper, an approach is proposed for correcting article errors in English translation results in order to improve the performance of a MT system. We check the article and the singular/plural form of the headword in a NP at the same time. This is different from most of early researches in which only articles are considered. Our correcting algorithm is based on simple, viable n-gram model whose parameters can be obtained using the WWW search engine Google. Using much less features than those used in the early researches, we experimentally showed that our approach could perform the promising results with a precision of 86.2% on all classes of article errors.**

I. INTRODUCTION

Recently, Machine Translation (MT) has made significant achievements. However its quality has been known to be very poor, and has not reached users' satisfactory level. With the rapid progress of Internet technology, international conversation becomes more frequent. The necessary of improvement of its performance proves to be even more serious. In [1], Schafer established a special error typology after gathering examples of errors typically corrected by post-editors in their various MT projects. In their classification, the translation error classification includes lexical errors (e.g., terminology and idioms), syntactic errors (e.g., word order and clause analysis) and grammatical errors (e.g., article and tense).

In this paper, we address on the problem how to automatically correct article selection errors and the singular/plural form errors of the headword modified by this article in order to improve the performance of MT systems. The reason that we process the two kinds of errors together is that how to use an article is largely depended on the singular/plural form of the headword in this phrase. It certainly seems to be more reasonable for us to check article and the singular/plural form of the headword together. For example, for the translation result "#[1]It could be a growing pain", our research goal is to automatically correct the phrase "#a growing pain" to "growing pains", for *growing pains* is a plural only compound noun which always occurs in plural form and there should be no article in this sentence either. In the following discussion, we call these kinds of errors as "article errors" which include both article selection errors and singular/plural form errors.

[1] "#" means that the text is grammatically wrong.

The article selection is to decide when to use a (an), the, or zero articles at the beginning of a noun phrase (NP), in which the singular/plural form of the headword also need to be determined accordingly. It is one of the most complex problems in translation result generation in MT. Especially when source texts are written in some languages, such as Chinese and Japanese, which do not have any articles or mark the countability, the problem becomes more difficult.

In this paper, after some discussion of the related research (section 2), we introduce an approach to correct article errors. In our correction algorithm, simple and viable n-gram-based model is proposed to select the correct target NP from candidates. The parameters of the model (web counts of queries) can be obtained with the help of WWW search engine Google (section 3). We evaluated the correction ability of our approach for all article error classes on the revised test set. In order to learn the performance on each error class separately, we made up several special test sets using artificial errors and corrected sentences to realize the evaluation. We show that our algorithm performs the promising results (section 4). Finally the paper ends with some conclusions and future work (section 5).

II. RELATED RESEARCH

Most of the early researches on article problem have been using rules to improve the quality of text [2, 3]. In the research of Japanese and English translation, Heine [3] focused on Japanese NP and classified whether it is definite or indefinite. The detection rules were all extracted by hands. The result that 79.5% of the NPs were classified with an accuracy of 98.9% was reported.

However writing a complete article rule set is a time-consuming work which also needs the help of linguists. Contrasting to these hand-made rules, Knight and Chander in their work [4] proposed an automatic post-editor to insert articles into English. They looked article selection as a classification problem, and trained a decision-tree builder on 4,000,000 NPs from *Wall Street Journal* text to learn whether to generate *the* or *a/an*. They used a variety of lexical (e.g., words before or after the article), syntactic (e.g., POS), and semantic (e.g., tense) features to generate over 200,000 rules automatically. They achieved an overall accuracy of 78%.

The cases of zero article that Knight and Chander did not consider was further extended by Minnen et al. in their research [5]. Some additional features were added such as the head of the NP, the presence of a determiner in the NP and

the countability of the head. They used a larger feature set than that of Knight and Chander [4]. Instead of decision-tree, they used memory-based learner to train and test their model. Their approach performed better than that of [4] and achieved an accuracy of 83.6%.

Lee [6] applied a log-linear model to automatically restore missing articles based on the features which were similar to those employed in [5]. However the model that they applied used the maximum entropy property to estimate the conditional probabilities of each article feature. They reported an accuracy of 87.7%.

III. Our Approach

From the discussion above, we can find that most of the researches viewed article selection problem as a classification problem whose input is a large set of features drawn from the context of a NP and whose output is the most likely article for that NP. It is obvious that they all have assumed that the context of any article is correct without mistakes. Yet it is not always the case. For text translated by MT, we cannot be sure that all the other parts of translated result are correct except the article. Because of the consideration, we take article and the singular/plural form of the headword together into account in our work.

A. Article Error Classes

Article error classes in MT are designed to make us aware of the main types of errors that can occur when using MT. After gathering examples of article errors from translation text, we put these article errors into 3 classes: *loss*, *unwanted* and *misuse*. *Loss* class includes the article errors missing *a/an*, *the* and *s* for the plural form of the headword (e.g., #He is teacher. #they are boy). *Unwanted* class includes the article errors that use *a/an the* and *s* to the headword when they are unnecessary (e.g., #The doctors and nurses should care for patients). *Misuse* class includes the article errors using an article or *s* when another form is needed (e.g., #It could be a growing pain. "a" should be replaced by adding "s" to "pain").

We analyzed 300 English sentences with article errors (437 errors). These sentences had been translated from 150 Chinese sentences and 150 Japanese sentences by three MT systems among which *Babelfish* [7] and *Infoseek* [8] are two on-line MTs and *Kingsoft* is a MT software. In Table I, we show the distribution of article error classes, which can provide an overview of the necessary corrections and the enhancements to be carried out in the corresponding MT systems. In this table, "-the" means "the" is required; "+a/an" means "a/an" is unnecessary; "the->s" means that "the" should be replaced by adding "s" to headword.

TABLE I
Article Error Distribution

Classes		Number	Ratio (%)
Loss	- a/an	107	24.5
	- the	12	2.7
	- s	163	37.3
Unwanted	+ a/an	23	5.3
	+ the	56	12.8
	+ s	29	6.6
Misuse	a/an -> s	17	3.9
	the -> a	19	4.4
	the -> s	11	2.5
Total		437	100

From the distribution, we can find out that *loss* class appears most frequently, and the next is *unwanted* class. Both of them have a large proportion about 90% of the total errors.

B. Article Forms

In the related research discussed above, they all have classified articles in 3 classes, *a/an, the* and *zero*. In this paper, we also consider the singular/plural form of the headword at the same time. Therefore we put articles into 5 forms. The detail is shown in Table II. "0" stands for *zero* article.

TABLE II
Article Forms

	a/an	the	0
singular	C_1	C_2	C_3
plural		C_4	C_5

For example "a student" belongs to C_1 and "the post offices" belongs to C_4. Though "a/an... plural" might occur in English writings, but we are sure that this form cannot be seen in translation results. Therefore we do not take this case into account.

C. Web-based Model

The Internet is a rich source of data for natural language processing. In our research, we view WWW as a large expression dictionary and assume that the article form (C_i, $1 \le i \le 5$) with largest occurrence probability is most likely correct given a certain context. We use (1) as follow to describe it.

$$C = \arg\max \Pr(Ci \mid context) \quad (1 \le i \le 5) \tag{1}$$

$\Pr(Ci \mid context)$ is the occurrence probability of C_i when given the *context*. We then use Maximum Likelihood Estimation (MLE) for a conversion from probability to frequency calculating.

$$\Pr(Ci \mid context) \approx \frac{\Pr(Ci, context)}{\Pr(context)} = \frac{frequency(Ci, context)}{frequency(context)} \tag{2}$$

With the help of a WWW search engine we can obtain web frequency values (web counts of queries). We describe *context* using a dyad *(l, r)*. It means *context* consists of *l* words left to the article and *r* words right to the headword. *frequency*(*Ci*,*context*) is the frequency of an article form co-occurring with the *context*. For an article form given a certain *(l, r)*, the n-gram is queried by Google "literal query", which uses the quoted n-gram directly as a search term for exactly

match. *frequency*(*context*) is the frequency of a certain context, which is estimated by Google "* query" using "*" operator. "*" can stand for any word in a search term. Google search also supports the Boolean operator. To retrieve pages that include either word A or word B, use an uppercase OR between terms. In our searching, we introduce this operator "OR" to expand a search term into all its morphological forms for more accurate estimation.

For example "She cut the apple in two", when we would like to estimated the frequency of "the apple" given the context *(1,1)*, we query "cut the apple in" using Google and obtain its frequency of 541. The frequency of the context is 2,460,000 estimated by the query "cut * * in", in which " * * " stands for any two possible words. We also can apply "OR" to expand "cut" for two other morphological forms: "cutting" and "cuts". Using "(cut OR cutting OR cuts)" instead of "cut" for queries in some cases, we can get more accurate estimation.

Querying in WWW adds noise to the data, we certainly lose some precision compared to supervised statistical models, but we assume that the size of the WWW will compensate the rough queries. Keller and Lapata [9] showed the evidence of the reliability of the web counts for natural language processing. Although they also experimentally showed that web-based approach could overcome data sparseness for bi-grams in [10], the problem still exists in our experiments. When the web count returned is zero, we smooth zero by adding it to 0.01.

Although we used Google as our search engine, we did not use Google Web API service for programme realization in our later experiments, for Google limits to 1000 automated queries per day. As we just need web counts returned for each search query, we directly extracted these numbers from the web pages found.

D. *Processing Flow*

Fig. 1 gives an overview of our processing flow for correcting the article errors.

For a translated English sentence, we firstly need to extract the NP in it (e.g., "a student" in "I am a student") to process in the further using. We use Apple Pie Parser [11], which is a bottom-up probabilistic chart parser. It finds the parse tree with the best score by best-first search algorithm and has been said to perform well for noun phrases especially.

The next step, we classify the extracted NP, and generate the other article candidates. For example, "a student" belongs to C_1, and then we generate "the student" (C_2), "student" (C_3), "the students" (C_4) and "students" (C_5) as article candidates.

We pluralize nouns based on general rules. For example, words that end with *-ch, x, s* or *s* require an *-es* for the plural. Compound nouns create some problems when we need to pluralize them. Since no real rules exist for how to pluralize all the words, we summarized from "*Guide to English Grammar and Writing*" [12] and processed our experimental data following the rules below.

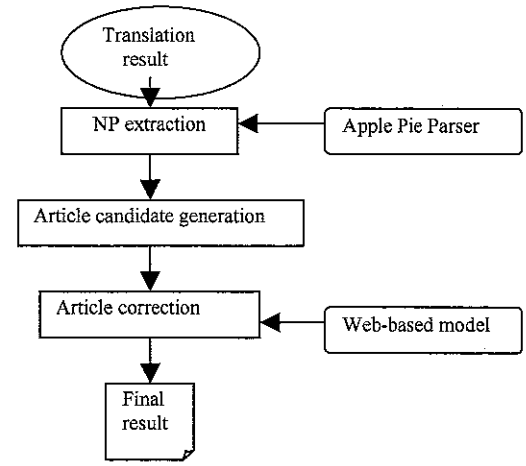

Fig.1. Flow of Article Correction Processing

1. Pluralize the last word of a compound noun (e.g., bedrooms, film stars).
2. When "*woman*" or "*man*" are the modifiers in a compound noun, pluralize both of the words (e.g., Women-drivers).
3. When a compound noun is made up as "noun + preposition (or prep. phrase)", pluralize the noun (e.g., fathers-in-law).
4. When the compound noun is made up as "verb (or past participle) + adverb", pluralize the last word (e.g., grown-ups, stand-bys).

Then using the web-based model explained in section 3.3, we can get final corrected result that is with largest probability estimated by this model.

E. *An Example*

For a better understanding of our approach, we give an example to show how it works. Suppose that we have the following translation sentence that we want to correct article error if any. We assume that there are no other errors in the sentence.

"I have consented on the conditions that he should pay beforehand."

Using Apple Pie Parser, we can extract the NP " the conditions". Then we generate the other article classes, "a condition", "the condition", "condition", "conditions" for article candidates.

Suppose the context is *(2, 1)*. For "a condition", "the condition" and "the conditions", these NPs have 2 words, we generate the query "(consented OR consent OR consents) on * * that" for them while use "(consented OR consent OR

TABLE III

OCCURRENCE PROBABILITY OF THE CANDIDATES

Class	Context	(C_i,context)	P(C_i|context)
C_1	4,700	1	0.0002
C_2	4,700	384	0.082
C_3	879	833	0.95
C_4	4,700	4	0.0008
C_5	879	2	0.0023

consents) on * that" for the left NPs to estimate *frequency*(*context*) . Replace "*" with the NPs for *frequency*(*Ci*,*context*) and calculate the final occurrence probability of each candidate according to Formula (2).

From the data in Table III, we can find out that "condition" most likely occurs in the sentence with the largest occurrence probability of 0.95. The final sentence is corrected as the following:

"I have consented on condition that he should pay beforehand."

One thing that we need to mention here is that the second and third columns in Table 3 were the web counts returned by Google in June 2005. The numbers found for the same query might be a little different as time goes on, because the number of web pages has been increasing day by day. However this slight instability has little effect on our approach.

IV. EXPERIMENTS AND RESULTS

We devised the experiments for some intentions. Besides evaluating the correction ability of our approach, we also want to learn the performance of our approach on different article error classes respectively. Discussed in section 3.1, for *loss* and *unwanted* classes appear most frequently, we need to focus on the two necessary corrections for the enhancements in the corresponding MT systems.

$$\mathrm{Re}call = \frac{A}{AB} \tag{3}$$

$$\mathrm{Pr}ecision = \frac{A}{AC} \tag{4}$$

$$F-score = \frac{2\times \mathrm{Pr}ecision\times \mathrm{Re}call}{\mathrm{Pr}ecision+\mathrm{Re}call} \tag{5}$$

We use Recall and Precision to evaluate the performance of our correction approach. The fundamental Recall/Precision definition is adapted to IE system evaluation. We borrowed the measures using the following definition for our evaluation. In these formulas, A stands for the number of article errors corrected accurately; AB stands for the total number of article errors; AC stands for the total number of article errors corrected. The Recall and Precision are defined in (3) and (4). We also introduced F-score when we need to consider the Recall and Precision at the same time, and in the paper, F-score is calculated according to (5).

A. Experiment Based on Tagged Data

We selected 1,000 sentences as the test set from bilingual parallel texts in English readings. In the test set, the 700 Chinese sentences and 300 Japanese sentences were extracted from "New Concept English Book III" and "English Journal" (May and June 2005 editions) respectively. These sentences were then translated into English by three MT systems: *Babelfish* [7], *Infoseek* [8] and *Kingsoft*. An English teacher who is a native speaker corrected the other translation errors (e.g., tense, word order) based on the original English texts. He kept only articles and singular/plural forms of headwords unchanged and just provided correct article forms for our later evaluation experiments. The average length of these English sentences is 12.7 words and they have 1,237 articles in total.

We evaluated the performances of our approach when given 4 different context: *(1, 1)* (the context is one word left to the NP and one word right to it), *(2, 0)*, *(2, 1)* and *(2, 2)*. Table IV shows the results.

From the results, we can find out that we achieve the best accuracy of 86.2% given the context *(2, 1)*. Though we have not yet done any comparison experiments on the same test data with the literature discussed in section 2, we can achieve such promising accuracy based on the simple and unsupervised model. It used just only context frequency and co-occurrence frequency as features much less than those used in the models proposed in the literature.

B. Addition Test Set With Artificial Errors and corrected sentences

In order to learn the performance of our approach on different article error classes respectively, we need special test sets for these error classes separately. We made up the test sets using artificial errors and corrected sentences.

For the 1000 test sentences used in section 4.1, we corrected the left of article errors manually and insured that all the sentences were correct without any error. We made up artificial errors just by replacing the correctly used articles with one of the others or changing the singular/plural form of headwords.

For example, suppose that we have a correct sentence "This is the nicest song I have ever heard". We want to rewrite it and let it have a certain error "-the" of the *loss* class. We then delete "the" and put the sentences "This is nicest song I have ever heard" to the test set of "-the" errors.

Table V shows the correction accuracy in *loss* and *unwanted* classes (for the two are a large proportion of the

TABLE IV

CORRENTION ACCURACY

Context	Recall (%)	Precision (%)	F-score (%)
(1, 1)	82.2	85.7	83.9
(2, 0)	79.1	83.6	81.3
(2, 1)	83.5	86.2	84.8
(2, 2)	72.8	79.1	75.8

TABLE V

CORRENTION ACCURACY FOR ERROR CLASSES

Class	Recall (%)	Precision (%)	F-score (%)
- a/an	87.7	88.9	88.3
- the	88.7	90.4	89.5
- s	82.5	86.7	84.5
+ a/an	81.1	85.8	83.4
+ the	78.8	82.3	80.5
+ s	79.1	89.2	83.8

total article errors). The correcting precisions of "-a/an", "-the" and "-s" are 88.9%, 90.4% and 86.7% respectively, and the total precision of *loss* class is 88.7% while the precision of *unwanted* class is 85.8%.

From the results we can find out that our approach performs much better in *loss* errors than in *unwanted* errors. This maybe can be explained for some reasons. The context of NPs in which an article is really necessary has much closer relationship with the NPs so that it is more effective to be used in our algorithm. For *unwanted* error class which also occurs frequently, we would like to define some general rules in the further to detect the situations of using an article when there should be none. For example, some countable nouns of institutions are used without articles (e.g., he is in church /college /jail /class), but the rules could be more complex than that.

V. CONCLUSION

In this paper, we explained how to correct article errors in English translation outputs in order to improve the performance of MT systems. Different from the early researches in which only article is considered while other parts are assumed to be correct, we check the article and the singular/plural form of the headword in a NP together. We proposed an unsupervised web-based model whose parameters can be obtained using the WWW search engine Google. We experimentally showed that our approach based on this simple model could perform the promising results with a precision of 86.2% on all article error classes, and 88.7% on *loss* error class.

As part of our future work, we would also like to combine some general rules with our approach to improve the correct rate.

Using WWW is an exciting direction for NLP, but the web-based methods invariably introduce noise in the resulting frequency data. How to eliminate noise data is the key to improve web-based methods. Our next step is aiming at evaluating the Internet resource by distinguishing the useful and noise data.

REFERENCES

[1] F. SCHÄFER, "MT-post-editing: How to shed light on the "unknown task" - experiments made at SAP", Controlled Language Application Workshop (CLAW-03), page: 133-140. 2003.

[2] F.Bond and S. Ikehara, "When and how to disambiguate?-countability in machine translation", In International Seminar on Multimodal Interactive Disambiguation: MIDDM-96, page, 149-160.1996.

[3] J.Heine, "Definiteness predictions for Japanese noun phrase", In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistic: COLING/ACL-98, page, 519-525. 1998.

[4] K. Knight, and I. Chander, "Automated postediting of documents", In Proceedings of the Twelfth National Conference on Artificial Intelligence. AAAI Press, 1994.

[5] G. Minnen, F. Bond, and A. Copestake, "Memory-based learning for article generation", In Proceedings of the 4th Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop, page: 43-48. 1998.

[6] J. Lee, "Automatic article restoration", In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, page: 31-36. 2004.

[7] Babelfish. http://babelfish.altavista.com/

[8] Infoseek. http://www.infoseek.co.jp/Honyaku?pg= honyaku_top.Html &svp =SEEK

[9] F. Keller, and M. Lapata, "Using the web to obtain frequencies for unseen bigrams", Computational Linguistics 29, 3, 459-484. 2003.

[10] [F. Keller, M. Lapata, and O. Ourioupina, "Using the web to overcome data sparseness", In Proceedings of the Conference on Empirical Methods in Natural Language Processing. page: 230-237. 2002.

[11] Apple Pie Parser. http://nlp.cs.nyu.edu/app/

[12] Guide to English Grammar and Writing. http:// cctc. commnet. edu /grammar/