Effects of Empty Categories on Machine Translation

Tagyoung Chung and **Daniel Gildea**
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

We examine effects that empty categories have on machine translation. Empty categories are elements in parse trees that lack corresponding overt surface forms (words) such as dropped pronouns and markers for control constructions. We start by training machine translation systems with manually inserted empty elements. We find that inclusion of some empty categories in training data improves the translation result. We expand the experiment by automatically inserting these elements into a larger data set using various methods and training on the modified corpus. We show that even when automatic prediction of null elements is not highly accurate, it nevertheless improves the end translation result.

1 Introduction

An empty category is an element in a parse tree that does not have a corresponding surface word. They include traces such as Wh-traces which indicate movement operations in interrogative sentences and dropped pronouns which indicate omission of pronouns in places where pronouns are normally expected. Many treebanks include empty nodes in parse trees to represent non-local dependencies or dropped elements. Examples of the former include traces such as relative clause markers in the Penn Treebank (Bies et al., 1995). An example of the latter include dropped pronouns in the Korean Treebank (Han and Ryu, 2005) and the Chinese Treebank (Xue and Xia, 2000).

In languages such as Chinese, Japanese, and Korean, pronouns are frequently or regularly dropped when they are pragmatically inferable. These languages are called *pro-drop* languages. Dropped pronouns are quite a common phenomenon in these languages. In the Chinese Treebank, they occur once in every four sentences on average. In Korean the Treebank, they are even more frequent, occurring in almost every sentence on average. Translating these pro-drop languages into languages such as English where pronouns are regularly retained could be problematic because English pronouns have to be generated from nothing.

There are several different strategies to counter this problem. A special NULL word is typically used when learning word alignment (Brown et al., 1993). Words that have non-existent counterparts can be aligned to the NULL word. In phrase-based translation, the phrase learning system may be able to learn pronouns as a part of larger phrases. If the learned phrases include pronouns on the target side that are dropped from source side, the system may be able to insert pronouns even when they are missing from the source language. This is an often observed phenomenon in phrase-based translation systems. Explicit insertion of missing words can also be included in syntax-based translation models (Yamada and Knight, 2001). For the closely related problem of inserting grammatical function particles in English-to-Korean and English-to-Japanese machine translation, Hong et al. (2009) and Isozaki et al. (2010) employ preprocessing techniques to add special symbols to the English source text.

In this paper, we examine a strategy of automatically inserting two types of empty elements from the Korean and Chinese treebanks as a preprocess-

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 636–645,
MIT, Massachusetts, USA, 9-11 October 2010.

Korean		
T	0.47	trace of movement
(NP *pro*)	0.88	dropped subject or object
(WHNP *op*)	0.40	empty operator in relative constructions
?	0.006	verb deletion, VP ellipsis, and others

Chinese		
(XP (-NONE- *T*))	0.54	trace of A'-movement
(NP (-NONE- *))	0.003	trace of A-movement
(NP (-NONE- *pro*))	0.27	dropped subject or object
(NP (-NONE- *PRO*))	0.31	control structures
(WHNP (-NONE- *OP*))	0.53	empty operator in relative constructions
(XP (-NONE- *RNR*))	0.026	right node raising
(XP (-NONE- *?*))	0	others

Table 1: List of empty categories in the Korean Treebank (top) and the Chinese Treebank (bottom) and their per-sentence frequencies in the training data of initial experiments.

		BLEU
Chi-Eng	No null elements	19.31
	w/ *pro*	19.68
	w/ *PRO*	19.54
	w/ *pro* and *PRO*	20.20
	w/ all null elements	**20.48**
Kor-Eng	No null elements	20.10
	w/ *pro*	**20.37**
	w/ all null elements	19.71

Table 2: BLEU score result of initial experiments. Each experiment has different empty categories added in. *PRO* stands for the empty category used to mark control structures and *pro* indicates dropped pronouns for both Chinese and Korean.

ing step. We first describe our experiments with data that have been annotated with empty categories, focusing on zero pronouns and traces such as those used in control constructions. We use these annotations to insert empty elements in a corpus and train a machine translation system to see if they improve translation results. Then, we illustrate different methods we have devised to automatically insert empty elements to corpus. Finally, we describe our experiments with training machine translation systems with corpora that are automatically augmented with empty elements. We conclude this paper by discussing possible improvements to the different methods we describe in this paper.

2 Initial experiments

2.1 Setup

We start by testing the plausibility of our idea of preprocessing corpus to insert empty categories with ideal datasets. The Chinese Treebank (LDC2005T01U01) is annotated with null elements and a portion of the Chinese Treebank has been translated into English (LDC2007T02). The Korean Treebank version 1.0 (LDC2002T26) is also annotated with null elements and includes an English translation. We extract null elements along with tree terminals (words) and train a simple phrase-based machine translation system. Both datasets have about 5K sentences and 80% of the data was used for training, 10% for development, and 10% for testing.

We used Moses (Koehn et al., 2007) to train machine translation systems. Default parameters were used for all experiments. The same number of GIZA++ (Och and Ney, 2003) iterations were used for all experiments. Minimum error rate training (Och, 2003) was run on each system afterwards, and the BLEU score (Papineni et al., 2002) was calculated on the test sets.

There are several different empty categories in the different treebanks. We have experimented with leaving in and out different empty categories for different experiments to see their effect. We hypothesized that nominal phrasal empty categories such as dropped pronouns may be more useful than other ones, since they are the ones that may be missing in the source language (Chinese and Korean) but have counterparts in the target (English). Table 1 summarizes empty categories in Chinese and Korean treebank and their frequencies in the training data.

2.2 Results

Table 2 summarizes our findings. It is clear that not all elements improve translation results when included in the training data. For the Chinese to English experiment, empty categories that mark control structures (*PRO*), which serve as the subject of a dependent clause, and dropped pronouns (*pro*), which mark omission of pragmatically in-

word	$P(e \mid *\text{pro}*)$	word	$P(e \mid *\text{PRO}*)$
the	0.18	to	0.45
i	0.13	NULL	0.10
it	0.08	the	0.02
to	0.08	of	0.02
they	0.05	as	0.02

Table 3: A lexical translation table from the Korean-English translation system (left) and a lexical translation from the Chinese-English translation system (right). For the Korean-English lexical translation table, the left column is English words that are aligned to a dropped pronoun (*pro*) and the right column is the conditional probability of $P(e \mid *\text{pro}*)$. For the Chinese-English lexical translation table, the left column is English words that are aligned to a control construction marker (*PRO*) and the right column is the conditional probability of $P(e \mid *\text{PRO}*)$.

ferable pronouns, helped to improve translation results the most. For the Korean to English experiment, the dropped pronoun is the only empty category that seems to improve translation.

For the Korean to English experiment, we also tried annotating whether the dropped pronouns are a subject, an object, or a complement using information from the Treebank's function tags, since English pronouns are inflected according to case. However, this did not yield a very different result and in fact was slightly worse. This is possibly due to data sparsity created when dropped pronouns are annotated. Dropped pronouns in subject position were the overwhelming majority (91%), and there were too few dropped pronouns in object position to learn good parameters.

2.3 Analysis

Table 3 and Table 4 give us a glimpse of why having these empty categories may lead to better translation. Table 3 is the lexical translation table for the dropped pronoun (*pro*) from the Korean to English experiment and the marker for control constructions (*PRO*) from the Chinese to English experiment. For the dropped pronoun in the Korean to English experiment, although there are errors, the table largely reflects expected translations of a dropped pronoun. It is possible that the system is inserting pronouns in right places that would be missing otherwise. For the control construction marker in the Chinese to English experiment, the top translation for *PRO* is the English word *to*, which is expected since Chinese clauses that have control construction markers often translate to English as to-infinitives. However, as we discuss in the next paragraph, the presence of control construction markers may affect translation results in more subtle ways when combined with phrase learning.

Table 4 shows how translations from the system trained with null elements and the system trained without null elements differ. The results are taken from the test set and show extracts from larger sentences. Chinese verbs that follow the empty node for control constructions (*PRO*) are generally translated to English as a verb in to-infinitive form, a gerund, or a nominalized verb. The translation results show that the system trained with this null element (*PRO*) translates verbs that follow the null element largely in such a manner. However, it may not be always closest to the reference. It is exemplified by the translation of one phrase.

Experiments in this section showed that preprocessing the corpus to include some empty elements can improve translation results. We also identified which empty categories maybe helpful for improving translation for different language pairs. In the next section, we focus on how we add these elements automatically to a corpus that is not annotated with empty elements for the purpose of preprocessing corpus for machine translation.

3 Recovering empty nodes

There are a few previous works that have attempted restore empty nodes for parse trees using the Penn English Treebank. Johnson (2002) uses rather simple pattern matching to restore empty categories as well as their co-indexed antecedents with surprisingly good accuracy. Gabbard et al. (2006) present a more sophisticated algorithm that tries to recover empty categories in several steps. In each step, one or more empty categories are restored using patterns or classifiers (five maximum-entropy and two perceptron-based classifiers to be exact).

What we are trying to achieve has obvious similarity to these previous works. However, there are several differences. First, we deal with different languages. Second, we are only trying to recover

Chinese	English Reference	System trained w/ nulls	System trained w/o nulls
PRO 贯彻	implementing	implementation	implemented
PRO 逐步 形成	have gradually formed	to gradually form	gradually formed
PRO 吸引 外资 作为	attracting foreign investment	attracting foreign investment	attract foreign capital

Table 4: The first column is a Chinese word or a phrase that immediately follows empty node marker for Chinese control constructions. The second column is the English reference translation. The third column is the translation output from the system that is trained with the empty categories added in. The fourth column is the translation output from the system trained without the empty categories added, which was given the test set without the empty categories. Words or phrases and their translations presented in the table are part of larger sentences.

a couple of empty categories that would help machine translation. Third, we are not interested in recovering antecedents. The linguistic differences and the empty categories we are interested in recovering made the task much harder than it is for English. We will discuss this in more detail later.

From this section on, we will discuss only Chinese-English translation because Chinese presents a much more interesting case, since we need to recover two different empty categories that are very similarly distributed. Data availability was also a consideration since much larger datasets (bilingual and monolingual) are available for Chinese. The Korean Treebank has only about 5K sentences, whereas the version of Chinese Treebank we used includes 28K sentences.

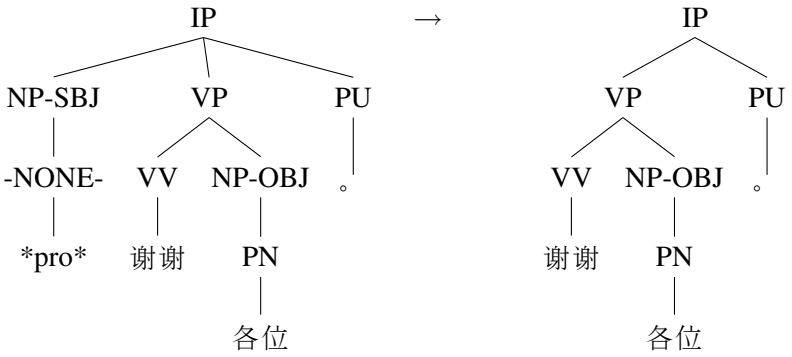
The Chinese Treebank was used for all experiments that are mentioned in the rest of this Section. Roughly 90% of the data was used for the training set, and the rest was used for the test set. As we have discussed in Section 2, we are interested in recovering dropped pronouns (*pro*) and control construction markers (*PRO*). We have tried three different relatively simple methods so that recovering empty elements would not require any special infrastructure.

3.1 Pattern matching

Johnson (2002) defines a pattern for empty node recovery to be a minimally connected tree fragment containing an empty node and all nodes co-indexed with it. Figure 1 shows an example of a pattern. We extracted patterns according this definition, and it became immediately clear that the same definition that worked for English will not work for Chinese. Table 5 shows the top five patterns that match control constructions (*PRO*) and dropped pronouns (*pro*). The top pattern that matches *pro* and *PRO* are both exactly the same, since the pattern will be matched against parse trees where empty nodes have been deleted.

When it became apparent that we cannot use the same definition of patterns to successfully restore empty categories, we added more context to the patterns. Patterns needed more context for them to be able to disambiguate between sites that need to be inserted with *pro*s and sites that need to be inserted with *PRO*s. Instead of using minimal tree fragments that matched empty categories, we included the parent and siblings of the minimal tree fragment in the pattern (pattern matching method 1). This way, we gained more context. However, as can be seen in Table 5, there is still a lot of overlap between patterns for the two empty categories. However, it is more apparent that at least we can choose the pattern that will maximize matches for one empty category and then discard that pattern for the other empty category.

We also tried giving patterns even more context by including terminals if preterminals are present in the pattern (pattern matching method 2). In this way, we are able have more context for patterns such as (VP VV (IP (NP (-NONE- *PRO*)) VP)) by knowing what the verb that precedes the empty category is. Instead of the original pattern, we would have patterns such as (VP (VV 决 定) (IP (NP (-NONE- *PRO*)) VP)). We are able to gain more context because some verbs select for a control construction. The Chinese verb 决定 generally translates to English as *to decide* and is more often followed by a control construction than by a dropped pronoun. Whereas the pattern (VP (VV 决定) (IP (NP (-NONE- *PRO*)) VP)) occurred 154 times in the training data, the pattern (VP (VV 决定) (IP (NP (-NONE- *pro*)) VP)) occurred only 8 times in the



(IP (NP-SBJ (-NONE- *pro*)) VP PU) (IP VP PU)

Figure 1: An example of a tree with an empty node (left), the tree stripped of an empty node (right), and a pattern that matches the example. Sentences are parsed without empty nodes and if a tree fragment (IP VP PU) is encountered in a parse tree, the empty node may be inserted according to the learned pattern (IP (NP-SBJ (-NONE- *pro*)) VP PU).

PRO		*pro*	
Count	Pattern	Count	Pattern
12269	(IP (NP (-NONE- *PRO*)) VP)	10073	(IP (NP (-NONE- *pro*)) VP)
102	(IP PU (NP (-NONE- *PRO*)) VP PU)	657	(IP (NP (-NONE- *pro*)) VP PU)
14	(IP (NP (-NONE- *PRO*)) VP PRN)	415	(IP ADVP (NP (-NONE- *pro*)) VP)
13	(IP NP (NP (-NONE- *PRO*)) VP)	322	(IP NP (NP (-NONE- *pro*)) VP)
12	(CP (NP (-NONE- *PRO*)) CP)	164	(IP PP PU (NP (-NONE- *pro*)) VP)

PRO		*pro*	
Count	Pattern	Count	Pattern
2991	(VP VV NP (IP (NP (-NONE- *PRO*)) VP))	1782	(CP (IP (NP (-NONE- *pro*)) VP) DEC)
2955	(VP VV (IP (NP (-NONE- *PRO*)) VP))	1007	(VP VV (IP (NP (-NONE- *pro*)) VP))
850	(CP (IP (NP (-NONE- *PRO*)) VP) DEC)	702	(LCP (IP (NP (-NONE- *pro*)) VP) LC)
765	(PP P (IP (NP (-NONE- *PRO*)) VP))	684	(IP IP PU (IP (NP (-NONE- *pro*)) VP) PU)
654	(LCP (IP (NP (-NONE- *PRO*)) VP) LC)	654	(TOP (IP (NP (-NONE- *pro*)) VP PU))

Table 5: Top five minimally connected patterns that match *pro* and *PRO* (top). Patterns that match both *pro* and *PRO* are shaded with the same color. The table on the bottom show more refined patterns that are given added context by including the parent and siblings to minimally connected patterns. Many patterns still match both *pro* and *PRO* but there is a lesser degree of overlap.

training data.

After the patterns are extracted, we performed pruning similar to the pruning that was done by Johnson (2002). The patterns that have less than 50% chance of matching are discarded. For example, if (IP VP) occurs one hundred times in a treebank that is stripped of empty nodes and if pattern (IP (NP (-NONE- *PRO*)) VP) occurs less than fifty times in the same treebank that is annotated with empty nodes, it is discarded.[1] We also found that we can discard patterns that occur very rarely (that occur only once) without losing much accuracy. In cases where there was an overlap between two empty categories, the pattern was chosen for either *pro* or *PRO*, whichever that maximized the number of matchings and then discarded for the other.

3.2 Conditional random field

We tried building a simple conditional random field (Lafferty et al., 2001) to predict null elements. The model examines each and every word boundary and decides whether to leave it as it is, insert *pro*, or insert *PRO*. The obvious disadvantage of this method is that if there are two consecutive null elements, it will miss at least one of them. Although there were some cases like this in the treebank, they were rare enough that we decided to ignore them. We first tried using only differently sized local windows of words as features (CRF model 1). We also experimented with adding the part-of-speech tags of words as features (CRF model 2). Finally, we experimented with a variation where the model is given each word and its part-of-speech tag and its immediate parent node as features (CRF model 3).

We experimented with using different regularizations and different values for regularizations but it did not make much difference in the final results. The numbers we report later used L_2 regularization.

3.3 Parsing

In this approach, we annotated nonterminal symbols in the treebank to include information about empty categories and then extracted a context free grammar from the modified treebank. We parsed with the modified grammar, and then deterministically recovered the empty categories from the trees. Figure 2 illustrates how the trees were modified. For every empty node, the most immediate ancestor of the empty node that has more than one child was annotated with information about the empty node, and the empty node was deleted. We annotated whether the deleted empty node was *pro* or *PRO* and where it was deleted. Adding where the child was necessary because, even though most empty nodes are the first child, there are many exceptions.

We first extracted a plain context free grammar after modifying the trees and used the modified grammar to parse the test set and then tried to recover the empty elements. This approach did not work well. We then applied the latent annotation learning procedures of Petrov et al. (2006)[2] to refine the nonterminals in the modified grammar. This has been shown to help parsing in many different situations. Although the state splitting procedure is designed to maximize the likelihood of of the parse trees, rather than specifically to predict the empty nodes, learning a refined grammar over modified trees was also effective in helping to predict empty nodes. Table 6 shows the dramatic improvement after each split, merge, and smoothing cycle. The gains leveled off after the sixth iteration and the sixth order grammar was used to run later experiments.

	PRO			*pro*		
Cycle	Prec.	Rec.	F1	Prec	Rec.	F1
1	0.38	0.08	0.13	0.38	0.08	0.12
2	0.52	0.23	0.31	0.37	0.18	0.24
3	0.59	0.46	0.52	0.43	0.24	0.31
4	**0.62**	0.50	0.56	**0.47**	0.25	0.33
5	0.61	0.52	**0.56**	0.47	0.33	0.39
6	0.60	**0.53**	0.56	0.46	0.39	**0.42**
7	0.58	0.52	0.55	0.43	**0.40**	0.41

Table 6: Result using the grammars output by the Berkeley state-splitting grammar trainer to predict empty categories

3.4 Results

Table 7 shows the results of our experiments. The numbers are very low when compared to accuracy reported in other works that were mentioned in the beginning of this Section, which dealt with the Penn English Treebank. Dropped pronouns are especially

[1] See Johnson (2002) for more details.

[2] http://code.google.com/p/berkeleyparser/

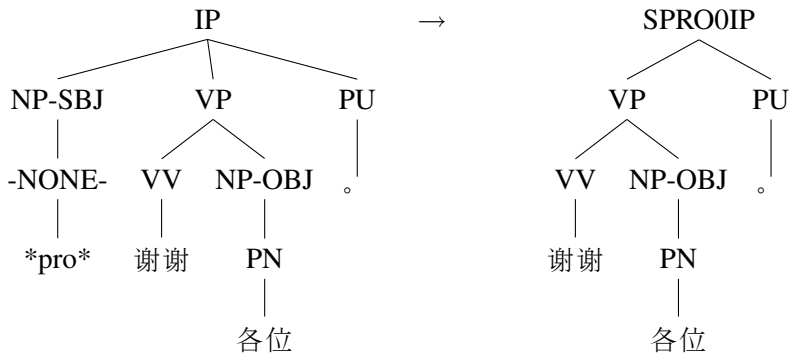


Figure 2: An example of tree modification

	PRO			*pro*		
	Prec.	Rec.	F1	Prec	Rec.	F1
Pattern 1	0.65	**0.61**	**0.63**	0.41	0.23	0.29
Pattern 2	0.67	0.58	0.62	0.46	0.24	0.31
CRF 1	0.66	0.31	0.43	0.53	0.24	0.33
CRF 2	**0.68**	0.46	0.55	**0.58**	0.35	**0.44**
CRF 3	0.63	0.47	0.54	0.54	0.36	0.43
Parsing	0.60	0.53	0.56	0.46	**0.39**	0.42

Table 7: Result of recovering empty nodes

hard to recover. However, we are dealing with a different language and different kinds of empty categories. Empty categories recovered this way may still help translation. In the next section, we take the best variation of the each method use it to add empty categories to a training corpus and train machine translation systems to see whether having empty categories can help improve translation in more realistic situations.

3.5 Analysis

The results reveal many interesting aspects about recovering empty categories. The results suggest that tree structures are important features for finding sites where markers for control constructions (*PRO*) have been deleted. The method utilizing patterns that have more information about tree structure of these sites performed better than other methods. The fact that the method using parsing was better at predicting *PRO*s than the methods that used the conditional random fields also corroborates this finding. For predicting dropped pronouns, the method using the CRFs did better than the others. This suggests that rather than tree structure, local context of words and part-of-speech tags maybe more important features for predicting dropped pronouns. It may also suggest that methods using robust machine learning techniques are better outfitted for predicting dropped pronouns.

It is interesting to note how effective the parser was at predicting empty categories. The method using the parser requires the least amount of supervision. The method using CRFs requires feature design, and the method that uses patterns needs human decisions on what the patterns should be and pruning criteria. There is also room for improvement. The split-merge cycles learn grammars that produce better parse trees rather than grammars that predict empty categories more accurately. By modifying this learning process, we may be able to learn grammars that are better suited for predicting empty categories.

4 Experiments

4.1 Setup

For Chinese-English, we used a subset of FBIS newswire data consisting of about 2M words and 60K sentences on the English side. For our development set and test set, we had about 1000 sentences each with 10 reference translations taken from the NIST 2002 MT evaluation. All Chinese data was re-segmented with the CRF-based Stanford Chinese segmenter (Chang et al., 2008) that is trained on the segmentation of the Chinese Treebank for consistency. The parser used in Section 3 was used to parse the training data so that null elements could be recovered from the trees. The same method for recovering null elements was applied to the train-

	BLEU	BP	*PRO*	*pro*
Baseline	23.73	1.000		
Pattern	23.99	0.998	0.62	0.31
CRF	**24.69***	1.000	0.55	0.44
Parsing	23.99	1.000	0.56	0.42

Table 8: Final BLEU score result. The asterisk indicates statistical significance at $p < 0.05$ with 1000 iterations of paired bootstrap resampling. BP stands for the brevity penalty in BLEU. F1 scores for recovering empty categories are repeated here for comparison.

ing, development, and test sets to insert empty nodes for each experiment. The baseline system was also trained using the raw data.

We used Moses (Koehn et al., 2007) to train machine translation systems. Default parameters were used for all experiments. The same number of GIZA++ (Och and Ney, 2003) iterations were used for all experiments. Minimum error rate training (Och, 2003) was run on each system afterwards and the BLEU score (Papineni et al., 2002) was calculated on the test set.

4.2 Results

Table 8 summarizes our results. Generally, all systems produced BLEU scores that are better than the baseline, but the best BLEU score came from the system that used the CRF for null element insertion. The machine translation system that used training data from the method that was overall the best in predicting empty elements performed the best. The improvement is 0.96 points in BLEU score, which represents statistical significance at $p < 0.002$ based on 1000 iterations of paired bootstrap resampling (Koehn, 2004). Brevity penalties applied for calculating BLEU scores are presented to demonstrate that the baseline system is not penalized for producing shorter sentences compared other systems.[3]

The BLEU scores presented in Table 8 represent the best variations of each method we have tried for recovering empty elements. Although the difference was small, when the F1 score were same for two variations of a method, it seemed that we could get slightly better BLEU score with the variation that had higher recall for recovering empty elements rather the variation with higher precision.

We tried a variation of the experiment where the CRF method is used to recover *pro* and the pattern matching is used to recover *PRO*, since these represent the best methods for recovering the respective empty categories. However, it was not as successful as we thought would be. The resulting BLEU score from the experiment was 24.24, which is lower than the one that used the CRF method to recover both *pro* and *PRO*. The problem was we used a very naïve method of resolving conflict between two different methods. The CRF method identified 17463 sites in the training data where *pro* should be added. Of these sites, the pattern matching method guessed 2695 sites should be inserted with *PRO* rather than *pro*, which represent more than 15% of total sites that the CRF method decided to insert *pro*. In the aforementioned experiment, wherever there was a conflict, both *pro* and *PRO* were inserted. This probably lead the experiment to have worse result than using only the one best method. This experiment suggest that more sophisticated methods should be considered when resolving conflicts created by using heterogeneous methods to recover different empty categories.

Table 9 shows five example translations of source sentences in the test set that have one of the empty categories. Since empty categories have been automatically inserted, they are not always in the correct places. The table includes the translation results from the baseline system where the training and test sets did not have empty categories and the translation results from the system (the one that used the CRF) that is trained on an automatically augmented corpus and given the automatically augmented test set.

5 Conclusion

In this paper, we have showed that adding some empty elements can help building machine translation systems. We showed that we can still benefit from augmenting the training corpus with empty elements even when empty element prediction is less than what would be conventionally considered robust.

We have also shown that there is a lot of room for improvement. More comprehensive and sophisti-

[3] We thank an anonymous reviewer for tipping us to examine the brevity penalty.

source	中国 计划 *PRO* 投资 在 基础 设施 上
reference	china plans to invest in the infrastructure
system trained w/ nulls	china plans to invest in infrastructure
system trained w/o nulls	china 's investment in infrastructure
source	有利 *PRO* 巩固 香港 的 贸易 和 航运 中心
reference	good for consolidating the trade and shipping center of hong kong
system trained w/ nulls	favorable to the consolidation of the trade and shipping center in hong kong
system trained w/o nulls	hong kong will consolidate the trade and shipping center
source	一些 大型 企业 *PRO* 逐步 走向 破产
reference	some large - sized enterprises to gradually go bankrupt
system trained w/ nulls	some large enterprises to gradually becoming bankrupt
system trained w/o nulls	some large enterprises gradually becoming bankrupt
source	*pro* 目前 还 不 清楚
reference	it is not clear now
system trained w/ nulls	it is also not clear
system trained w/o nulls	he is not clear
source	*pro* 现在 还 不 清楚
reference	it is not clear yet
system trained w/ nulls	it is still not clear
system trained w/o nulls	is still not clear

Table 9: Sample translations. The system trained without nulls is the baseline system where the training corpus and test corpus did not have empty categories. The system trained with nulls is the system trained with the training corpus and the test corpus that have been automatically augmented with empty categories. All examples are part of longer sentences.

cated methods, perhaps resembling the work of Gabbard et al. (2006) may be necessary for more accurate recovery of empty elements. We can also consider simpler methods where different algorithms are used for recovering different empty elements, in which case, we need to be careful about how recovering different empty elements could interact with each other as exemplified by our discussion of the pattern matching algorithm in Section 3 and our experiment presented in Section 4.2.

There are several other issues we may consider when recovering empty categories that are missing in the target language. We only considered empty categories that are present in treebanks. However, there might be some empty elements which are not annotated but nevertheless helpful for improving machine translation. As always, preprocessing the corpus to address a certain problem in machine translation is less principled than tackling the problem head on by integrating it into the machine translation system itself. It may be beneficial to include consideration for empty elements in the decoding process, so that it can benefit from interacting with other elements of the machine translation system.

Acknowledgments We thank the anonymous reviewers for their helpful comments. This work was supported by NSF grants IIS-0546554 and IIS-0910611.

References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for treebank II style. Penn Treebank Project, January.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Pi-Chuan Chang, Michel Galley, and Christopher Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232.

Ryan Gabbard, Seth Kulick, and Mitchell Marcus. 2006. Fully parsing the Penn Treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 184–191, New York

City, USA, June. Association for Computational Linguistics.

Na-Rae Han and Shijong Ryu. 2005. Guidelines for Penn Korean Treebank version 2.0. Technical report, IRCS, University of Pennsylvania.

Gumwon Hong, Seung-Wook Lee, and Hae-Chang Rim. 2009. Bridging morpho-syntactic gap between source and target sentences for English-Korean statistical machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 233–236.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics*, pages 244–251.

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, Demonstration Session*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain, July.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Machine Learning: Proceedings of the Eighteenth International Conference (ICML 2001)*, Stanford, California.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41th Annual Conference of the Association for Computational Linguistics (ACL-03)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

Nianwen Xue and Fei Xia. 2000. The bracketing guidelines for the Penn Chinese Treebank. Technical Report IRCS-00-08, IRCS, University of Pennsylvania.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Conference of the Association for Computational Linguistics (ACL-01)*, Toulouse, France.