Generating Case Markers in Machine Translation

Kristina Toutanova **Hisami Suzuki**
Microsoft Research
One Microsoft Way, Redmond WA 98052 USA
{hisamis,kristout}@microsoft.com

Abstract

We study the use of rich syntax-based statistical models for generating grammatical case for the purpose of machine translation from a language which does not indicate case explicitly (English) to a language with a rich system of surface case markers (Japanese). We propose an extension of *n*-best re-ranking as a method of integrating such models into a statistical MT system and show that this method substantially outperforms standard *n*-best re-ranking. Our best performing model achieves a statistically significant improvement over the baseline MT system according to the BLEU metric. Human evaluation also confirms the results.

1 Introduction

Generation of grammatical elements such as inflectional endings and case markers is an important component technology for machine translation (MT). Statistical machine translation (SMT) systems, however, have not yet successfully incorporated components that generate grammatical elements in the target language. Most state-of-the-art SMT systems treat grammatical elements in exactly the same way as content words, and rely on general-purpose phrasal translations and target language models to generate these elements (e.g., Och and Ney, 2002; Koehn et al., 2003; Quirk et al., 2005; Chiang, 2005; Galley et al., 2006). However, since these grammatical elements in the target language often correspond to long-range dependencies and/or do not have any words corresponding in the source, they may be difficult to model, and the output of an SMT system is often ungrammatical.

For example, Figure 1 shows an output from our baseline English-to-Japanese SMT system on a sentence from a computer domain. The SMT system, trained on this domain, produces a natural lexical translation for the English word *patch* as *correction program*, and translates *replace* into passive voice, which is more appropriate in Japanese.[1] However, there is a problem in the case marker assignment: the accusative marker *wo*, which was output by the SMT system, is completely inappropriate when the main verb is passive. This type of mistake in case marker assignment is by no means isolated in our SMT system: a manual analysis showed that 16 out of 100 translations had mistakes solely in the assignment of case markers. A better model of case assignment could therefore improve the quality of an SMT system significantly.

S: The patch replaces the .dll file.
O: 修正プログラムを.dll ファイルが置き換えられます。
shuusei puroguramu-wo .dll fairu-ga okikae-raremasu
correction program-ACC dll file-NOM replace-PASS
C: 修正プログラムで.dll ファイルが置き換えられます。
shuusei puroguramu-de .dll fairu-ga okikae-raremasu
correction program-with dll file-NOM replace-PASS

Figure 1: Example of SMT (S: source; O: output of MT; C: correct translation)

In this paper, we explore the use of a statistical model for case marker generation in English-to-Japanese SMT. Though we focus on the generation of case markers in this paper, there are many other surface grammatical phenomena that can be modeled in a similar way, so any SMT system dealing with morpho-syntactically divergent language pairs may benefit from a similar approach to modeling grammatical elements. Our model uses a rich set of syntactic features of both the source (English) and the target (Japanese) sentences, using context which is broader than that utilized by existing SMT systems. We show that the use of such features results in very high case assignment quality and also leads to a notable improvement in MT quality.

Previous work has discussed the building of special-purpose classifiers which generate grammatical elements such as prepositions (Hajič et al. 2002), determiners (Knight and Chander, 1994) and case markers (Suzuki and Toutanova, 2006) with an eye toward improving MT output. How-

[1] There is a strong tendency to avoid transitive sentences with an inanimate subject in Japanese.

Proceedings of NAACL HLT 2007, pages 49–56,
Rochester, NY, April 2007.

ever, these components have not actually been integrated in an MT system. To our knowledge, this is the first work to integrate a grammatical element production model in an SMT system and to evaluate its impact in the context of end-to-end MT.

A common approach of integrating new models with a statistical MT system is to add them as new feature functions which are used in decoding or in models which re-rank *n*-best lists from the MT system (Och et al., 2004). In this paper we propose an extension of the *n*-best re-ranking approach, where we expand *n*-best candidate lists with multiple case assignment variations, and define new feature functions on this expanded candidate set. We show that expanding the *n*-best lists significantly outperforms standard *n*-best re-ranking. We also show that integrating our case prediction model improves the quality of translation according to BLEU (Papineni et al., 2002) and human evaluation.

2 Background

In this section, we provide necessary background of the current work.

2.1 Task of case marker prediction

Our definition of the case marker prediction task follows Suzuki and Toutanova (2006). That is, we assume that we are given a source English sentence, and its translation in Japanese which does not include case markers. Our task is to predict all case markers in the Japanese sentence.

We determine the location of case marker insertion using the notion of *bunsetsu*. A bunsetsu consists of one content (head) word followed by any number of function words. We can therefore segment any sentence into a sequence of bunsetsu by using a part-of-speech (POS) tagger.

Once a sentence is segmented into bunsetsu, it is trivial to determine the location of case markers in a sentence: each bunsetsu can have at most one case marker, and the position of the case maker within a phrase is predictable, i.e., the rightmost position before any punctuation marks. The sentence in Figure 1 thus has the following bunsetsu analysis (denoted by square brackets), with the locations of potential case marker insertion indicated by □:

[修正'correction'□][プログラム'program'□][.dll□][ファイル'file'□][置き換えられます'replace-PASS'□。]

For each of these positions, our task is to predict the case marker or to predict NONE, which means that the phrase does not have a case marker.

case markers		grammatical functions	+*wa*
が	*ga*	subject; object	
を	*wo*	object; path	
の	*no*	genitive; subject	
に	*ni*	dative object, location	✓
から	*kara*	source	✓
と	*to*	quotative, reciprocal, *as*	✓
で	*de*	location,instrument, cause	✓
へ	*e*	goal, direction	✓
まで	*made*	goal (up to, until)	✓
より	*yori*	source, comparison target	✓
は	*wa*	Topic	

Table 1. Case markers to be predicted

The case markers we used for the prediction task are the same as those defined in Suzuki and Toutatnova (2006), and are summarized in Table 1: in addition to the case markers in a strict sense, the topic marker *wa* is also included as well as the combination of a case marker plus the topic marker for the case markers with the column +*wa* checked in the table. In total, there are 18 case markers to predict: ten simple case markers, the topic marker *wa*, and seven case+*wa* combinations. The case prediction task is therefore a 19-fold classification task: for each phrase, we assign one of the 18 case markers or NONE.

2.2 Treelet translation system

We constructed and evaluated our case prediction model in the context of a treelet-based translation system, described in Quirk et al. (2005).[2] In this approach, translation is guided by treelet translation pairs, where a treelet is a connected subgraph of a dependency tree.

A sentence is translated in the treelet system as follows. The input sentence is first parsed into a dependency structure, which is then partitioned into treelets, assuming a uniform probability distribution over all partitions. Each source treelet is then matched to a treelet translation pair, the collection of which will form the target translation. The target language treelets are then joined to form a single tree, and the ordering of all the nodes is determined, using the method described in Quirk et al. (2005).

Translations are scored according to a linear combination of feature functions:

$$score(t) = \sum_j \lambda_j f_j(t) \qquad (1)$$

[2] Though this paper reports results in the context of a treelet system, the model is also applicable to other syntax-based or phrase-based SMT systems.

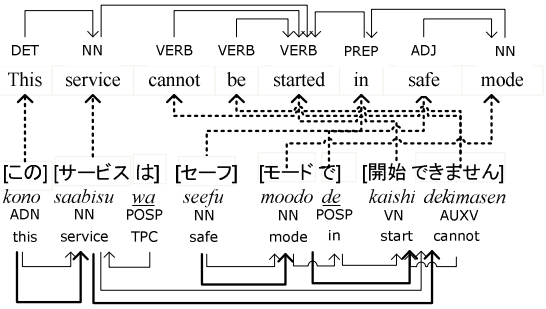

Figure 2. Aligned English-Japanese sentence pair

where λ_j are the model parameters and $f_j(t)$ is the value of the feature function j on the candidate t. There are ten feature functions in the treelet system, including log-probabilities according to inverted and direct channel models estimated by relative frequency, lexical weighting channel models following Vogel et al. (2003), a trigram target language model, an order model, word count, phrase count, average phrase size functions, and whole-sentence IBM Model 1 log-probabilities in both directions (Och et al. 2004). The weights of these models are determined using the max-BLEU method described in Och (2003). As we describe in Section 4, the case prediction model is integrated into the system as an additional feature function.

The treelet translation model is estimated using a parallel corpus. First, the corpus is word-aligned using GIZA++ (Och and Ney, 2000); then the source sentences are parsed into a dependency structure, and the dependency is projected onto the target side following the heuristics described in Quirk et al. (2005). Figure 2 shows an example of an aligned sentence pair: on the source (English) side, POS tags and word dependency structure are assigned (solid arcs); the word alignments between English and Japanese words are indicated by the dotted lines. On the target (Japanese) side, projected word dependencies (solid arcs) are available. Additional annotations in Figure 2, namely the POS tags and the bunsetsu dependency structure (bold arcs) on the target side, are derived from the treelet system to be used for building a case prediction model, which we describe in Section 3.

2.3 Data

All experiments reported in this paper are run using parallel data from a technical (computer) domain. We used two main data sets: train-500K, consisting of 500K sentence pairs which we used for training the baseline treelet system as well as the case prediction model, and a disjoint set of three data sets, lambda-1K, dev-1K and test-2K, which are used to integrate and evaluate the case prediction model in an end-to-end MT scenario. Some characteristics of these data sets are given in Table 2. We will refer to this table as we describe our experiments in later sections.

data set	# sent pairs	# of words (average sent length in words)	
		English	Japanese
train-500K	500K	7,909,198 (15.81)	9,379,240 (18.75)
lambda-1K	1,000	15,219(15.2)	20,660 (20.7)
dev-1K	1,000	15,397(15.4)	21,280 (21.3)
test-2K	2,000	30,198(15.1)	41,269 (20.6)

Table 2: Data set characteristics

3 Statistical Models for Case Prediction in MT

3.1 Case prediction model

Our model of case marker prediction closely follows our previous work of case prediction in a non-MT context (Suzuki and Toutanova, 2006). The model is a multi-class log-linear (maximum entropy) classifier using 19 classes (18 case markers and NONE). It assigns a probability distribution over case marker assignments given a source English sentence, all non-case marker words of a candidate Japanese translation, and additional annotation information. Let t denote a Japanese translation, s a corresponding source sentence, and A additional annotation information such as alignment, dependency structure, and POS tags (such as shown in Figure 2). Let $rest(t)$ denote the sequence of words in t excluding all case markers, and $case(t)$ a case marking assignment for all phrases in t. Our case marking model estimates the probability of a case assignment given all other information:

$$P_{case}\,(case\,(t) \mid rest\,(t), s, A)$$

The probability of a complete case assignment is a product over all phrases of the probability of the case marker of the phrase given all context features used by the model. Our model assumes that the case markers in a sentence are independent of each other given the input features. This independence assumption may seem strong, but the results presented in our previous work (Suzuki and Toutanova, 2006) showed that a joint model did not result in large improvements over a local one in predicting case markers in a non-MT context.

3.2 Model features and feature selection

The features of our model are similar to the ones described in Suzuki and Toutanova (2006). The main difference is that in the current model we applied a feature selection and induction algorithm to determine the most useful features and feature combinations. This is important for understanding what sources of information are important for predicting grammatical elements, but are currently absent from SMT systems. We used 490K sentence pairs for training the case prediction model, which is a subset of the train-500K set of Table 2. We divided the remaining 10K sentences for feature selection (5K-feat) and for evaluating the case prediction models on reference translations (5K-test, discussed in Section 3.3). The paired data is annotated using the treelet translation system: as shown in Figure 2, we have source and target word dependency structure, source language POS and word alignment directly from the aligned treelet structure. Additionally, we used a POS tagger of Japanese to assign POS to the target sentence as well as to parse the sentence into bunsetsu (indicated by brackets in Figure 2), using the method described in Section 2.1. We then compute bunsetsu dependency structure on the target side (indicated by bold arcs in Figure 2) based on the word dependency structure projected from English. We apply this procedure to annotate a paired corpus (in which case the Japanese sentence is a reference translation) as well as translations generated by the SMT system (which may potentially be ill-formed).

We derived a large set of possible features from these annotations. The features are represented as feature templates, such as "Headword POS=X", which generate a set of binary features corresponding to different instantiations of the template, such as "Headword POS=NOUN". We applied an automatic feature selection and induction algorithm to the base set of templates.

The feature selection algorithm considers the original templates as well as arbitrary (bigram and trigram) conjunctions of these templates. The algorithm performs forward stepwise feature selection, choosing templates which result in the highest increase in model accuracy on the 5K-feat set mentioned above. The algorithm is similar to the one described in McCallum (2003).

The application of this feature selection procedure gave us 17 templates, some of which are shown in Table 3, along with example instantiations for the phrase headed by *saabisu* 'service' from Figure 2. Conjunctions are indicated by &. Note that many features that refer to POS and syntactic (parent) information are selected, on both the target and source sides. We also note that the context required by these features is more extensive than what is usually available during decoding in an SMT system due to a limit imposed on the treelet or phrase size. For example, our model uses word lemma and POS tags of up to six words (previous word, next word, word in position +2, head word, previous head word and parent word), which covers more context than the treelet system we used (the system imposes the treelet size limit of four words). This means that the case model can make use of much richer information from both the source and target than the baseline MT system. Furthermore, our model makes better use of the context by combining the contributions of multiple sources of knowledge using a maximum entropy model, rather than using the relative frequency estimates with a very limited amount of smoothing, which are used by most state-of-the art SMT systems.

Features	Example
Words in position −1 and +2	*kono,moodo*
Headword & previous headword	*saabisu&kono*
Parent word	*kaishi*
Aligned word	*service*
Parent of word aligned to headword	*started*
Next word POS	*NOUN*
Next word & next word POS	*seefu&NN*
Headword POS	*NOUN*
Parent headword POS	*VN*
Aligned to parent word POS & next word POS & prev word POS	*VERB&NN&and*
Parent POS of word aligned to headword	*VERB*
Aligned word POS & headword POS & prev word POS	*NN&NN&ADN*
POS of word aligned to headword	*NOUN*

Table 3: Features for the case prediction model

3.3 Performance on reference translations

Before discussing the integration of the case prediction model with the MT system, we present an evaluation of the model on the task of predicting the case assignment of *reference* translations. This performance constitutes an upper bound on the model's performance in MT, because in reference translations, the word choice and the word order are perfect.

Table 4 summarizes the results of the reference experiments on the 5K-test set using two metrics: accuracy, which denotes the percentage of phrases for which the respective model guessed the case marker correctly, and BLEU score against the reference translation. For com-

Model	ACC	BLEU
Baseline (frequency)	58.9	40.0
Baseline (490K LM)	87.2	83.6
Log-linear model	94.9	93.0

Table 4: Accuracy (%) and BLEU score for case prediction when given correct context (reference translations) on the 5K-test set

parison, we also include results from two baselines: a frequency-based baseline, which always assigns the most likely class (NONE), and a language model (LM) baseline, which is one of the standard methods of generating grammatical elements in MT. We trained a word-trigram LM using the CMU toolkit (Clarkson and Rosenfeld, 1997) on the same 490K sentences which we used for training the case prediction model.

Table 4 shows that our model performs substantially better than both baselines: the accuracy of the frequency-based baseline is 59%, and an LM-based model improves it to 87.2%. In contrast, our model achieves an accuracy of 95%, which is a 60% error reduction over the LM baseline. It is also interesting to note that as the accuracy goes up, so does the BLEU score.

These results show that our best model can very effectively predict case markers when the input to the model is clean, i.e., when the input has correct words in correct order. Next, we see the impact of applying this model to improve MT output.

4 Integrating Case Prediction Models in MT

In the end-to-end MT scenario, we integrate our case assignment model with the SMT system and evaluate its contribution to the final MT output.

As a method of integration with the MT system, we chose an *n*-best re-ranking approach, where the baseline MT system is left unchanged and additional models are integrated in the form of feature functions via re-ranking of *n*-best lists from the system. Such an approach has been taken by Och et al. (2004) for integrating sophisticated syntax-informed models in a phrase-based SMT system. We also chose this approach for ease of implementation: as discussed in Section 3.2, the features we use in our case model extend over long distance, and are not readily available during decoding. Though a tighter integration with the decoding process is certainly worth exploring in the future, we have taken an approach here that allows fast experimentation.

Within the space of *n*-best re-ranking, we have considered two variations: the standard *n*-best re-ranking method, and our significantly better performing extension. These are now discussed in turn.

4.1 Method 1: Standard *n*-best re-ranking

This method is a straightforward application of the *n*-best re-ranking approach described in Och et al. (2004). As described in Section 2.2, our baseline SMT system is a linear model which weighs the values of ten feature functions. To integrate a case prediction model, we simply add it to the linear model as an 11th feature function, whose value is the log-probability of the case assignment of the candidate hypothesis *t* according to our model. The weights of all feature functions are then re-estimated using max-BLEU training on the *n*-best list of the lambda-1K set in Table 2. As we show in Section 5, this re-ranking method did not result in good performance.

4.2 Method 2: Re-ranking of expanded candidate lists

A drawback of the previous method is that in an *n*-best list, there may not be sufficiently many case assignment variations of existing hypotheses. If this is the case, the model cannot be effective in choosing a hypothesis with a good case assignment. We performed a simple experiment to test this. We took the first (best) hypothesis *t* from the MT system and generated the top 40 case variations *t'* of *t*, according to the case assignment model. These variations differ from *t* only in their case markers. We wanted to see what fraction of these new hypotheses *t'* occurred in a 1000-best list of the MT system. In the dev-1K set of Table 2, the fraction of new case variations of the first hypothesis occurring in the 1000-best list of hypotheses was 0.023. This means that only less than one (2.3% of 40 = 0.92) case variant of the first hypothesis is expected to be found in the 1000-best list, indicating that even an *n*-best list for a reasonably large *n* (such as 1000) does not contain enough candidates varying in case marker assignment.

In order to allow more case marking candidates to be considered, we propose the following method to expand the candidate translation list: for each translation *t* in the *n*-best list of the baseline SMT system, we also consider case assignment variations of *t*. For simplicity, we chose to consider the top *k* case assignment variations of each hypothesis according to our case model,[3] for $1 \leq k \leq 40$.[4]

[3] From a computational standpoint, it is non-trivial to con-

After we expand the translation candidate set, we compute feature functions for all candidates and train a linear model which chooses from this larger set. While some features (e.g., word count feature) are easy to recompute for a new candidate, other features (e.g., treelet phrase translation probability) are difficult to recompute. We have chosen to recompute only four features of the baseline model: the language model feature, the word count feature, and the direct and reverse whole-sentence IBM Model 1 features, assuming that the values of the other baseline model features for a casing variation *t'* of *t* are the same as their values for *t*. In addition, we added the following four feature functions, specifically meant to capture the extent to which the newly generated case marking variations differ from the original baseline system hypotheses they are derived from:

- **Generated:** a binary feature with a value of 0 for original baseline system candidates, and a value of 1 for newly generated candidates.
- **Number NONE→non-NONE:** the count of case markers changed from NONE to non-NONE with respect to an original translation candidate.
- **Number non-NONE→NONE:** the count of case markers changed from non-NONE to NONE.
- **Number non-NONE→non-NONE:** the count of case markers changed from non-NONE to another non-NONE case marker.

Note that these newly defined features all have a value of 0 for original baseline system candidates (i.e., when k=0) and therefore would have no effect in Method 1. Therefore, the only difference between our two methods of integration is the presence or absence of case-expanded candidate translations.

5 Experiments and Results

5.1 Data and settings

For our end-to-end MT experiments, we used three datasets in Table 2 that are disjoint from the train-500K data set. They consist of source English sentences and their top 1000 candidate translations produced by the baseline SMT system. These datasets are the lambda-1K set for training the weights λ of the linear model from Equation (1), the dev-1K set for model selection, and the test-2K set for final testing including human evaluation.

Models	#MT hypotheses	#case expansions	BLEU	Oracle BLEU
Baseline	1	0	37.99	37.99
Method 1	20	0	37.83	41.79
	100	0	38.02	42.79
	1000	0	38.08	43.14
Method 2	1	1	38.18	38.75
	1	10	38.42	40.51
	1	20	38.54	41.15
	1	40	38.41	41.74
Method 2	20	10	**38.91**	45.32
	20	20	38.72	45.94
	20	40	38.78	46.56
	100	10	38.73	46.87
	100	20	38.64	47.47
	100	40	38.74	47.96

Table 5. Results of end-to-end experiments on the dev-1K set

5.2 Results

The results for the end-to-end experiments on the dev-1K set are summarized in Table 5. The table is divided into four sections. The first section (row) shows the BLEU score of the baseline SMT system, which is equivalent to the 1-best re-ranking scenario with no case expansion. The BLEU score for the baseline was 37.99. In the table, we also show the oracle BLEU scores for each model, which are computed by greedily selecting the translation in the candidate list with the highest BLEU score.[5]

The second section of Table 5 corresponds to the results obtained by Method 1, i.e., the standard n-best re-ranking, for n = 20, 100, and 1000. Even though the oracle scores improve as n is increased, the actual performance improves only slightly. These results show that the strategy of only including the new information as features in a standard n-best re-ranking scenario does not lead to an improvement over the baseline.

In contrast, Method 2 obtains notable improvements over the baseline. Recall that we expand the n-best SMT candidates with their k-best case marking variations in this method, and re-

sider all possible case assignment variations of a hypothesis: even though the case assignment score for a sentence is locally decomposable, there are still global dependencies in the linear model from Equation (1) due to the reverse whole-sentence IBM model 1 score used as a feature function.

[4] Our results indicate that additional case variations would not be helpful.

[5] A modified version of BLEU was used to compute sentence-level BLEU in order to select the best hypothesis per sentence. The table shows corpus-level BLEU on the resulting set of translations.

train the model parameters on the resulting candidate lists. For the values n=1 and k=1 (which we refer to as 1best-1case), we observe a small BLEU gain of .19 over the baseline. Even though this is not a big improvement, it is still better than the improvement of standard n-best reranking with a 1000-best list. By considering more case marker variations (k = 10, 20 and 40), we are able to gain about a half BLEU point over the baseline. The fact that using more case variations performs better than using only the best case assignment candidate proposed by the case model suggests that the proposed approach, which integrates the case prediction model as a feature function and retrains the weights of the linear model, works better than using the case prediction model as a post-processor of the MT output.

The last section of the table explores combinations of the values for n and k. Considering 20 best SMT candidates and their top 10 case variations gave the highest BLEU score on the dev-1K set of 38.91, which is an 0.92 BLEU points improvement over the baseline. Considering more case variations (20 or 40), and more SMT candidates (100) resulted in a similar but slightly lower performance in BLEU. This is presumably because the case model does affect the choice of content words as well, but this influence is limited and can be best captured when using a small number (n=20) of baseline system candidates.

Based on these results on the dev-1K set, we chose the best model (i.e., 20-best-10case) and evaluated it on the test-2K set against the baseline. Using the pair-wise statistical test design described in Collins et al. (2005), the BLEU improvement (35.53 vs. 36.29) was statistically significant (p < .01) according to the Wilcoxon signed-rank test.

5.3 Human evaluation

These results demonstrate that the proposed model is effective at improving the translation quality according to the BLEU score. In this section, we report the results of human evaluation to ensure that the improvements in BLEU lead to better translations according to human evaluators.

We performed human evaluation on the 20best-10case (n=20, k=10) and 1best-40case (n=1, k=40) models against the baseline using our final test set, the test-2K data. The performance in BLEU of these models on the full test-2K data was 35.53 for the baseline, 36.09 for the 1best-40case model, and 36.29 for the 20best-10case model, respectively.

		Fluency			Adequacy		
		Annotator #1			Annotator #1		
		S	B	E	S	B	E
Anno-	S	**27**	1	8	**17**	0	9
tator	B	1	**9**	16	0	**9**	12
#2	E	7	4	**27**	9	8	**36**

Table 6. Results of human evaluation comparing 20best-10case vs. baseline. **S**: proposed system is better; **B**: baseline is better; **E**: of equal quality

In our human evaluation, two annotators were asked to evaluate a random set of 100 sentences for which the models being compared produced different translations. The judges were asked to compare two translations, the baseline output from the original SMT system and the output chosen by the system augmented with the case marker generation component. Each judge was asked to run two separate evaluations along different evaluation criteria. In the evaluation of *fluency*, the judges were asked to decide which translation is more readable/grammatical, ignoring the reference translation. In the evaluation of *adequacy*, they were asked to judge which translation more correctly reflects the meaning of the reference translation. In either setting, they were not given the source sentence.

Table 6 summarizes the results of the evaluation of the 20best-10case model. The table shows the results along two evaluation criteria separately, fluency on the left and adequacy on the right. The evaluation results of Annotator #1 are shown in the columns, while those of Annotator #2 are in the rows. Each grid in the table shows the number of sentences the annotators classified as the proposed system output better (S), the baseline system better (B) or the translations are of equal quality (E). Along the diagonal (in boldface) are the judgments that were agreed on by the two annotators: both annotators judged the output of the proposed system to be more fluent in 27 translations, less fluent in 9 translations; they judged that our system output was more adequate in 17 translations and less adequate in 9 translations. Our system output was thus judged better under both criteria, though according to a sign test, the improvement is statistically significant (p < .01) in fluency, but not in adequacy.

One of the reasons for this inconclusive result is that human evaluation may be very difficult and can be unreliable when evaluating very different translation candidates, which happens often when comparing the results of models that consider n-best candidates where n>1, as is the case with the 20best-10case model. In Table 6,

		Fluency			Adequacy		
		Annotator #1			Annotator #1		
		S	B	E	S	B	E
Anno-	S	**42**	0	9	**30**	1	9
tator	B	1	**0**	7	0	**9**	7
#2	E	7	2	**32**	9	3	**32**

Table 7. Results of human evaluation comparing 1best-40case vs. baseline

we can see that the raw agreement rate between the two annotators (i.e., number of agreed judgments over all judgments) is only 63% (27+9+27 /100) in fluency and 62% (17+9+36/100) in adequacy. We therefore performed an additional human evaluation where translations being compared differ only in case markers: the baseline vs. the 1best-40case model output. The results are shown in Table 7.

This evaluation has a higher rate of agreement, 74% for fluency and 71% for adequacy, indicating that comparing two translations that differ only minimally (i.e., in case markers) is more reliable. The improvements achieved by our model are statistically significant in both fluency and adequacy according to a sign test; in particular, it is remarkable that on 42 sentences, the judges agreed that our system was better in fluency, and there were no sentences on which the judges agreed that our system caused degradation. This means that the proposed system, when choosing among candidates differing only in case markers, can improve the quality of MT output in an extremely precise manner, i.e. making improvements without causing degradations.

6 Conclusion

We have described a method of using a case marker generation model to improve the quality of English-to-Japanese MT output. We have shown that the use of such a model contributes to improving MT output, both in BLEU and human evaluation. We have also proposed an extension of *n*-best re-ranking which significantly outperformed standard *n*-best re-ranking. This method should be generally applicable to integrating models which target specific phenomena in translation, and for which an extremely large *n*-best list would be needed to cover enough variants of the phenomena in question.

Our model improves the quality of generated case markers in an extremely precise manner. We believe this result is significant, as there are many phenomena in the target language of MT that may be improved by using special-purpose models, including the generation of articles, auxiliaries, inflection and agreement. We plan to extend and generalize the current approach to cover these phenomena in morphologically complex languages in general in the future.

References

Clarkson, P.R. and R. Rosenfeld. 1997. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *ESCA Eurospeech*, pp. 2007-2010.

Collins, M., P. Koehn and I. Kučerová. 2005. Clause Restructuring for Statistical Machine Translation. In *ACL*, pp.531-540.

Chiang, D. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *ACL*.

Galley, M., J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang and I. Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *ACL*.

Koehn, P., F. J. Och and D. Marcu. 2003. Statistical Phrase-based Translation. In *HLT-NAACL*.

Hajič, J., M. Čmejrek, B. Dorr, Y. Ding, J. Eisner, D. Gildea, T. Koo, K. Parton, G. Penn, D. Radev and O. Rambow. 2002. Natural Language Generation in the Context of Machine Translation. *Technical report, Center for Language and Speech Processing, Johns Hopkins University 2002 Summer Workshop Final Report.*

Knight, K. and I. Chander. 1994. Automatic Postediting of Documents. In *AAAI*.

McCallum, A. 2003. Efficiently inducing features of conditional random fields. *In UAI.*

Och, F. J. 2003. Minimum Error-rate Training for Statistical Machine Translation. In *ACL*.

Och, F. J. and H. Ney. 2000. Improved Statistical Alignment Models. In *ACL*.

Och, F. J. and H. Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *ACL 2002.*

Och, F. J., D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin and D. Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *NAACL*.

Papineni, K., S. Roukos, T. Ward and W.J. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*.

Quirk, C., A. Menezes and C. Cherry. 2005. Dependency Tree Translation: Syntactically Informed Phrasal SMT. In *ACL.*

Suzuki, H. and K. Toutanova. 2006. Learning to Predict Case Markers in Japanese. In *ACL-COLING*.

Vogel, S., Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao and A. Waibel. 2003. The CMU Statistical Machine Translation System. In *Proceedings of the MT Summit.*