Transfer rule generation for a Japanese-Hungarian MT system

VARGA István, YOKOYAMA Shoichi

Yamagata University, Graduate School of Science and Engineering
dyn36150@dip.yz.yamagata-u.ac.jp, yokoyama@yz.yamagata-u.ac.jp

Abstract

Our main objective is to produce low-cost translation resources for low-resourced languages; in this paper we present a simple transfer rule generation algorithm. Our method relies on a small bilingual corpus and a bilingual dictionary of the selected languages. Previous methods generally attempt to generate translation templates for each sentence pair; we concentrate on accurately inducing the most frequent rules between the two languages. These rules can extend from low level grammatical (conjugation, inflection) rules to sentence templates. We present the first results of our method with the Japanese-Hungarian language pair.

1 Introduction

Creating a set of transfer rules for a rule-based or pattern-based system could take many man-years of work (Prószéky and Tihanyi, 2002); we attempt to simplify this process by automatically generating these rules in form of *translation templates* and *grammatical rules*. As *grammatical rules*, we target word-level rule correspondences, such as inflection and conjugation rules across languages, particularly important with agglutinative languages. We generate this transfer rules using a small or medium sized parallel corpus.

This paper is structured as follows: first we discuss the most significant related studies, next we focus on the problems of current translation template generation methods, followed by a detailed description of our method. Finally we evaluate our method and conclude with our findings.

2 Related work

There are numerous researches considering translation templates, some of the most relevant ones are presented in this section. Similar fields include syntactic approaches to statistical machine translation; we describe the most significant findings.

There are numerous relatively successful examples of shallow translation template extraction methods for closely related languages (Altintas and Güvenir, 2003; Cicekli, 2005). The main disadvantage of these methods is the impossibility of re-implementation with distant or non-related languages, since they only handle languages that share a large number of features.

Initial in-depth structure alignment methods attempt to identify complex, hierarchical structures such as phrase structures (Kaji et al., 1992) or dependency structures (Watanabe et al., 2000). Other methods include the Translation Template Learner (TTL) algorithm, which analyzes similarities and differences between translation pairs (Cicekli and Güvenir, 2003). This method's accuracy is 77% *('at least one correct template is among the top 5 candidates')* with an English-Turkish artificially collected parallel corpus, or 91% with statistical refinements proposed by Öz and Cicekli (1998). Refinements such as of Ong et al. showed its applicability to other language pairs (English-Tagalog) as well (2007).

Most heuristic methods try to generate translation templates or sentence patterns from each sentence pair. Moreover, methods that attempt to generate rules that cover the entire sentence pair, notoriously fail in doing so, because most language pairs do not manifest a high degree if similarity in their syntax. With matching similar sub-trees, these methods estimate that the remaining, unmatched sub-trees are also equivalent, producing many erroneous, useless and even contradictory results. This is especially the case with idioms or distant languages.

As a possible solution to the drawbacks of the pure statistical machine translation (weak on re-ordering; lack of target language fluency), syntactic approaches were proposed that work with traditional statistic models, beginning from a syntax-based statistical machine translation (Yamada and Knight, 2001); multi-level syntactic translation rule generation methods and *string-based* systems (Galley et al., 2004; Galley et al., 2006); *tree-based* systems (Lin, 2004; Liu et al., 2006) and *forest-based* translation (Mi et al., 2008). The *forest-based* method is more refined than the tree-based method, since instead of the 1-best tree a *packed forest* is passed to the decoder. To decrease the size of the forest, a *forest pruning algorithm* is performed (Mi et al., 2008). These methods perform better than the non-statistical ones, but require large bilingual corpora.

However, most of the above mentioned methods are not applicable with small or medium sized corpora. Although the statistical methods are the best performing ones, for good efficiency they understandably require large amount of data, thus on small or medium sized corpora they do not perform well.

3 Proposed method

In order to achieve high precision, our method is not trying to extract rules one by one from each sentence pair. Instead, it analyzes all instances of a certain rule, attempting to extract the most frequent, and thus the most suitable transfer rule. During this process, it looks for the most general rule as possible, subcategorizing or exemplifying only when needed.

For high precision, the method follows a bottom-to-top mechanism, looking to identify not only general translation templates, but also *local rules*, or frequent sub-sequences of a certain pattern. With this method grammatical rules regarding inflections or conjugations can also be recognized.

3.1 Resource details

To generate the transfer rules, the proposed method uses a *parsed bilingual corpus* and a *bilingual dictionary*. For robustness our method's bilingual corpus requires a specific format. Since *phrase structure rules* provide with a relatively detailed syntax of a language, we opted for a variant of its bracketed description (Figure 1). We used the Hungarian *MetaMorph* (Prószéky and Novák, 2005) and Japanese *Cabocha* (Kudo and Matsumoto, 2000) parsers. The Hungarian parser's output matches the desired format. Regarding the

Japanese parser, first we adapted it to the grammar description of Yamada (Moriyama, 2000), so that instead of morphemes we can work with words. Secondly, we modified the parser's output method to fit our corpus's format.

(S (NP (Adj *colorless*)(NP (Adj *green*)(N<:pl> *ideas<idea>*)))
(VP (V *sleep*)(Adv *furiously*)))

Figure 1: Bracketing of *'Colorless green ideas sleep furiously'* with <optional additional information>

There is no known digital bilingual corpus for Hungarian and Japanese. There are a number of automated methods (Resnik and Smith, 2003) and manual ones (Varga et al., 2005) that target corpus acquisition, only the latter proved to be prolific with our language pair. We created 4 Hungarian-Japanese parallel corpora based on the source of information: *#1 software related documents, #2 translated literature from a third language, #3 directly translated literature, #4 language books*. Among these, corpus#1 had the lowest building cost, but it proved to be unsuitable, since it used the same grammatical structures without any variations and a large amount of segments of the so-called Hungarian and Japanese text was in fact in English. Corpus#2 and corpus#3 were also unsuitable, since most of the texts were paraphrases, rather than translations of each other; in case of corpus#2 the 1-to-1 alignments were only around 14%. With corpus#4 no alignment was necessary, since all data had to be typewritten, thus the cost proving to be the highest. However, no noise or paraphrase was observed in any of the sources. Although the sentences are short, it might be suitable for transfer rule extraction, since its data is grammatically rich and well prepared due to its initial educational purpose. We used the following resources:

- Kiss: Japán nyelvtani összefoglaló (2001);
- 今岡十一郎: ハンガリー語四週間, 大学書林 (1986);
- 岡本真理: らくらく旅のハンガリー語, 三修社 (2003);
- 早稲田みか: CD エクスプレス ハンガリー語, 白水社 (2005);
- 早稲田みか: ハンガリー語の入門, 白水社 (2001).

3.2 Transfer rule generation

The method itself is composed from two steps: first we generate the language models for each languages, next the rules itself are generated. Below is a detailed description of the method.

3.2.1 Language model generation

In this step we are looking to build the language model of the two languages. We compute every sentence in turn, separately saving every possible sub-tree. For example, in the case of our sample sentence *'Colorless green ideas sleep furiously'*, we have 9 sub-trees, for each one a rule is generated (Figure 2).

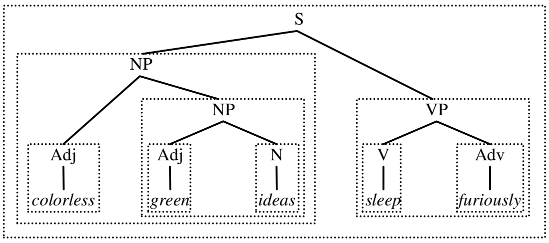

Figure 2: *'Colorless green ideas sleep furiously'* with its sub-trees

We can distinguish four types of rules:

(1) *head rule:* rules where the parent is the sentence itself. Each sentence has exactly one head rule (ex: *S→NP+VP*);

(2) *lexical rule:* rules whose children are *lexical categories (PoS)* (ex: *NP→Adj+N*);

(3) *terminal rule:* rules whose sole child is a *word*. The number of terminal rules is equal with the number of words that the sentence contains (ex: *V→sleep*);

(4) *regular rule:* every rule that is not head, lexical or terminal rule (ex: *NP→Adj+NP*).

During generation we count the frequency of each rule, also saving the sentence from which it was generated. Among these rules some of them are erroneous due to misanalyses of our parsers, but we do not perform any manual cleaning. Since in the next step we will work only with rules that have a certain frequency, we believe that erroneous rules will be ignored.

3.2.2 Transfer rule generation

In this step we are looking to build the language model of the two languages.

(a) Recursive search

We compute each head rule in descending order of their frequency. If there are children that are not *solved* (in case of head and general rules: a translation pattern or a grammatical rule already contains the children with their children), we move down to the children and attempt to compute the rules with the children and their children. There is a restraint on what we consider to be solved and what we attempt to solve. For example, in case of the rule $N_h(1)\rightarrow N_i(1)+...+N_{i+k}(1)$, the child $N_h(1)[\rightarrow N_j(1)+...+N_{j+k}(1)]$ is unsolved even if there is a similar solved node ($N_i(1)= N_i(2)$; $N_i(2)\rightarrow N_j(2)+...+N_{j+k}(2)$), if all their children are not the same.

(b) Solvable rule handling

If a rule that is investigated has only solved children or it is a lexical rule *($N\rightarrow LC_1(w_1)+LC_2(w_2)$)*, we look up all other rules that have the same parent-children configuration and we retrieve their corresponding sentences of the opposite language. Next, with the usage of the bilingual dictionary we attempt to identify to which sub-tree in the other language the lexical rule corresponds. We look up each lexical category's instance (word and stemmed expression, if it's available) from the lexical rule and mark the eventual correspondences. After all such correspondences are marked, we investigate the lowest level phrasal categories in the second language, counting how many identified instances it has in its sub-tree. The node or nodes with the maximum value are selected together with the sub-tree(s) as the possible transfer rule of the investigated initial lexical rule. Multiple transfer rules can be selected with this method (Figures 3 and 4).

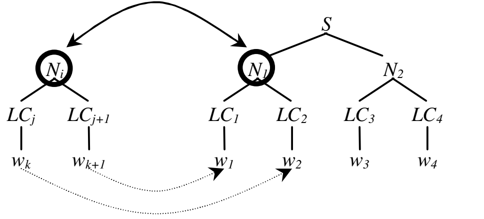

Figure 3: A single transfer rule candidate

For example, with the *$N_i\rightarrow LC_j(w_k)+LC_{j+1}(w_{k+1})$* rule, in case 1 the correspondences of w_k and w_{k+1} share the same lexical rule, thus N_1 is selected as a correspondence to N_i (Figure 3). In case 2 the correspondences are within different lexical rules, thus there are two lexical categories (LC_1 and LC_3) that share the maximum number of identified instances, even if this value is 1 (Figure 4). If no correspondences are found, no transfer rule candidate is returned.

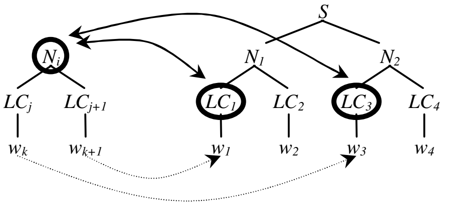

Figure 4: Multiple transfer rule candidates

(c) Instantiation

After all instances of the lexical rule are computed, a number of transfer rule candidates are gathered. In certain cases there could be instances of this rule's child whose translation was never retrieved. In this case the corresponding lexical category is *instantiated*, being replaced by its instance. For example, if our initial $N \rightarrow LC_1(w_1)+LC_2(w_2)$ rule's w_2 word did not have any correspondence, the rule becomes $N \rightarrow LC_1(w_1)+w_2$. Obviously we are going to have multiple new rules, depending on the number of the w_2 instances. For example, in case of the Japanese *PP→N+Part*, there is no general rule for a noun plus a particle, therefore the method correctly makes the judgment that the particle needs to be instantiated and new rules have to be generated for each instance (Table 1).

#	Japanese rule	Hungarian transfer rule candidate
1	*PP→N(あなた)+は*	*S→CONJ(és)+N<:sg>(ön)+PUNCT(?)*
2	*PP→N(これ)+は*	*VP→N<:sg>(mi)+N<:sg>(ez)*
3	*PP→N(あれ)+は*	*VP→N<:sg>(ki)+N<:sg>(az)*
4	*PP→N(あなた)+は*	*VP→N<:sg>(ön)+ ADJ(japán)*
5	PP→N(先生)+は	NP→DET(a)+N<:sg>(tanár)
6	PP→N(辞書)+は	NP→DET(a)+N<:sg>(szótár)
7	*PP→N(ハンガリー語)+は*	*NP→ADJ(magyar)+N<:sg>(nyelv)*

Table 1: Hungarian transfer rule candidates for PP→N+は (deleted candidates with italic)

(d) Noisy candidate elimination

If there are unmatched instances in the second language and their translation can not be found in the first language's rule, the transfer rule candidate is deleted. For example, none of the translations of *japán* (Japanese person; Japanese language) from example#4 could be found in the Japanese rule, thus the transfer rule was considered erroneous. On the other hand, the determinant *a* also doesn't have a translation in the Japanese rule, but it has no translation in the dictionary either (there is no corresponding Japanese translation), therefore it was allowed.

The remaining candidates are grouped by their common nodes and are saved with three values: *total number of candidates; total number of transfer rule instances; number of instances for the current rule*, along with marking the rule as *solved*. Since we do not use any thresholds within our method, these three numbers should indicate the confidence level of the transfer rule. We move back to the recursive search to investigate whether the parent rule can now be solved or not. For example, for *PP→N+は* only one transfer rules can be generated: *NP→DET+N<:sg>*. The corresponding values are (7, 2, 2).

4 Evaluation

For evaluation, we fragmented our corpus (#4) into 5 fragments. First we randomly separated 100 sentences, we used these as our evaluation data. Next we randomly separated 4 training corpora of 100, 500, 1000 and 2000 sentence pairs.

We performed automatic recall evaluation and a manual precision evaluation to validate our method. We used the rules whose *number of instances for the current rule* was at least 2.

4.1 Recall evaluation

We investigated to what percentage our method's output rules manage to cover the training data's phrase structure rules. We performed a weighted recall evaluation, weighting each rule by its frequency in the training corpus. Because of the *instantiation* feature many new rules are generated that are not part of the training data's phrase structure rules, during evaluation only we added these new rules to the training data.

We analyzed the Japanese coverage, separately evaluating the *head*, *general* and *lexical* rules. Lexical rules performed best, improving rapidly from 47.71% to 64.23% when the training data increased from 100 to 2000 sentence pairs (Figure 5). Head rules performed worst, with only up to a third of them managing to move up the parse trees all the way to the root.

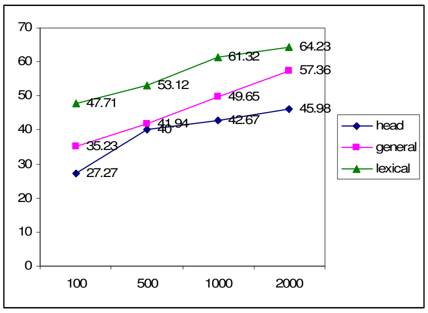

Figure 5: Weighted recall evaluation results

4.2 Precision evaluation

We manually simulated a machine translation system with 100 randomly selected Japanese sentences, by exchanging the templates with the corresponding words retrieved from the dictionary. We used a 5 to 1 scoring criteria, where 5 is a *perfect*, 1 is a *totally wrong* output sentence. We separately evaluated the *head*, *general* and *lexical* rules, but obviously their number was not equal. We performed the same evaluation on four training corpora: 100, 500, 1000 and 2000 samples.

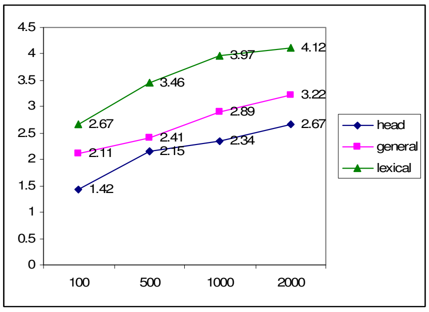

Figure 6: Precision evaluation results

Lexical items scored best, since understandably the errors on the lower level reflected within the general and head rules as well. However, with the increase in size of the training data, the accuracy of the lexical rules increased faster than the other two types of rules. We could not observe any major difference in behaviour between the general and head rules (Figure 6).

5 Discussions

Our method showed its biggest weakness during recall evaluation. Many rules could not be identified in the transfer rules, especially the ones which direct over larger sub-trees. There are two major reasons for this: linguistic differences and resource issues. Regarding precision, the once recalled rules showed a surprising accuracy, especially *lexical* ones. Precision problems can be mainly attributed to resource issues.

5.1 Linguistic differences

Our first observation is that the biggest reasons for the low recall value are the linguistic differences between Hungarian and Japanese. Syntax is different, sentence construction is also different; therefore one sub-tree in a certain language does not necessarily match another sub-tree in the other parse.

Other linguistic differences, such as expression of pronouns or the sentence topic manifest differently across languages, our method does not always generate the proper transfer rule in these cases. For example, with the *私+は (me, myself)* sub-tree rarely has any correspondence in Hungarian, since the agent (pronoun in this case) is expressed within the verb.

5.2 Resource issues

There are two types of resource issues. The first problem concerns the parsers and dictionary that we used. The parsers do not have a perfect accuracy, the noise produced by them reflected in the recall and accuracy results. The methodology of the parsers itself is different, with sub-trees not always matching a sub-tree in the other language, even when both parsers performed correctly. Our bilingual dictionary is noisy, no manual cleaning was performed in order to raise its recall or accuracy; many translations could not be identified.

The second problem concerns our corpus. Precision scores with smaller training data were low, because many erroneous transfer rules were generated besides the correct ones, but with the increase of the training data the frequency of correct rules also climbed rapidly. However, even with our biggest training data (2000 sentence pairs) the recall and precision values were not very high, but it is promising that from the second largest training data (1000 sentence pairs) the recall increase was between 3%-11%, the precision increase between 4%-13%. This significant increase shows that the problem does not lie in the method itself, the scores were low mostly because of the size of the corpus.

6 Conclusions

We presented a transfer rule generating method that uses a parsed bilingual corpus and a bilingual dictionary as resources. Although our biggest aim is low-cost in this research, during bilingual corpus acquisition we found ourselves in a contradictory situation: to generate low-cost transfer rules, we needed to manually create a small bilingual corpus. However, the cost of creating this corpus is insignificant when we think of the costs that a transfer rule system would require.

As a compromise between having a small or medium sized corpus with noisy parses and the desire to achieve a good performance, we did not concentrated on very specific or idiomatic expressions. As a result, with a small corpus we managed to achieve medium recall and good precision, with basic conjugation and inflectional rules being highly accurate. We showed that with the minimal increase in size of the bilingual corpus, overall precision, together with recall can quickly increase.

References

Altintas, K., Güvenir, H. A. (2003). Learning Translation Templates from Closely Related Languages, *KES 2003*, pp. 756-762.

Cicekli, I. (2005). Learning translation templates with type constrains, *MT Summit X, Proceedings of Second Workshop on Example-Based Machine Translation*, pp. 27-33.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. *Proceedings of COLING-ACL,* pp. 961-968.

Galley, M., Hopkins, M., Knight, K., Marcu, D. (2004). What's in a translation rule? *Proceedings of NAACL-HLT 2004,* pp. 273-280.

Kaji, H., Kida, Y., Morimoto, Y. (1992). Learning Translation Templates from Bilingual Texts, *Proceedings of COLING-92*, pp. 672-678.

Kudo, T., Matsumoto, Y. (2000). Japanese Dependency Structure Analysis using Cascaded Chunking, Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, pp. 18-25.

Lin, D. (2004). A path-based transfer model for machine translation, *Proceedings of the 20th COLING*.

Liu, Y., Huang, Y., Lin, S. (2006). Tree-to-string alignment template for statistical machine translation, *Proceedings of COLING-ACL,* pp. 609-616.

Mi, H., Huang, L., Liu, Q. (2008). Forest-Based Translation, *Proceedings of ACL-08: HLT,* pp. 192-199.

Moriyama, T. (2000). Japanese grammar starts here (ここからはじまる日本語文法), *Hituzi Syobo. (in Japanese)*

Ong, E., Go, K., Morga, M., Nunez, V., Veto, F. (2007). Extracting and Using Translation Templates in an Example-Based Machine Translation System, *Journal of Research in Science, Computing and Engineering* 3(4), pp. 83-100.

Öz, Z., Cicekli, I. (1998). Ordering Translation Templates by Assigning Confidence Factors, *Lecture Notes in Computer Science 1529*, Springer Verlag, pp. 51-61.

Prószéky, G, Tihanyi, L. (2002). MetaMorpho: A Pattern-Based Machine Translation System, *Proceedings of the 24th International Conference on Translating and the Computer*, London.

Resnik, P., Smith, N. A. (2003). The Web as a parallel corpus, *Computational Linguistics: special issue on web as corpus,* 29(3), pp. 349-380.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy, V. (2005). Parallel corpora for medium density languages, *Proceedings of the RANLP 2005 Conference*, pp. 590-596.

Varga, I, Yokoyama, S. (2007). Japanese-Hungarian Dictionary Generation using Ontology Resources, *In Proceedings of MT Summit XI*, pp. 483-490.

Watanabe, H., Kurohashi, S., Aramaki, E. (2000). Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation, *COLING-2000*.

Yamada, K., Knight, K. (2001). A syntax-based statistical translation model, *Proceedings of ACL*, pp. 523-530.