Transfer rule generation for a Japanese-Hungarian machine translation system

István Varga
Yamagata University,
4-3-16 Jonan, Yonezawa-shi Yamagata,
992-8510 Japan
dyn36150@ dip.yz.yamagata-u.ac.jp

Shoichi Yokoyama
Yamagata University,
4-3-16 Jonan, Yonezawa-shi Yamagata,
992-8510 Japan
yokoyama@yz.yamagata-u.ac.jp

Abstract

Rule based machine translation methods require a set of sophisticated transfer rules for good accuracy. To manually build such a bilingual resource, one requires many man-years of work performed by linguistic specialists. This cost is too high, especially in case of less represented language pairs, such as Hungarian and Japanese. This paper proposes a simple and robust method to automatically build a large coverage transfer rule set for the Hungarian-Japanese language pair. Our method uses a small parsed bilingual corpus and a bilingual dictionary of the selected languages. We concentrate on accurately inducing the most frequent target language translation rules from all instances of a source language rule. We achieved good accuracy especially for low level rules, which are especially important in case of agglutinative languages.

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 transfer rules, including word-level rule correspondences, such as inflection and conjugation rules. This is particularly crucial with agglutinative languages, such as Hungarian or Japanese. Both languages manifest a high degree of verb and adjective inflection, governed by grammatical rules for which bilingual transfer rule implementation can be very costly. Moreover, both languages present specific linguistic features that are again very costly to organize in a bilingual environment. For example, Hungarian has one of the most grammatical cases, estimated to be between 17 and 24 (László, 1977) (Table 1). In Japanese *particles* before the respective nouns are used instead of grammatical cases.

We attempt to generate the transfer rules using a small or medium sized parallel corpus and a bilingual dictionary, concentrating mainly on word-level, inflectional correspondences.

Grammatical case	**Code**	**Inflection**	**Example**
nominative	n	Ø	*ló*
accusative	a	-t	*lovat*
genitive	g	Ø, -nak/-nek	*lónak*
dative	d	-nak/-nek	*lónak*
instrumental	i	-val/-vel	*lóval*
illative	x	-ba/-be	*lóba*
inessive	2	-ban/-ben	*lóban*
elative	e	-ból/-ből	*lóból*
allative	t	-hoz/-hez/-höz	*lóhoz*
adessive	3	-nál/-nél	*lónál*
ablative	b	-tól/-től	*lótól*
sublative	s	-ra/-re	*lóra*
superessive	p	-n/-on/-en/-ön	*lovon*
delative	h	-ról/-ről	*lóról*
terminative	9	-ig	*lóig*
essive	w	-ul/-ül	*lóul*
(essive-)formal	f	-ként, -képp(en)	*lóként*
temporalis	m	-kor	*lókor*
causalis	c	-ért	*lóért*
sociative	q	-stul/-stül	*lovastul*
factive	y	-vá/-vé	*lóvá*
distributive	u	-nként	*lóként*
locativus	l	-tt; -n/-on/-en/-ön; -ben/-ban	*lóban*

Table 1: Hungarian grammatical cases of *ló* ('horse')

Because of the limited bilingual resources, our transfer rules will be incorporated into a rule based machine translation framework for assimilation purposes. Thus we target the most simple and general transfer rules that cover most of the language. In this stage we do not attempt to create rules for grammatical exceptions or idiomatic expressions.

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

2 Related work

There are numerous relatively successful examples of shallow translation template extraction methods for closely related languages (Altintas and Güvenir, 2003; Cicekli, 2005). 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; Öz and Cicekli, 1998; Ong et al., 2007). Most of these heuristic methods attempt to generate transfer rules from each sentence pair. With distant languages they notoriously fail, because after recognizing partial matches, these methods estimate that the remaining, unmatched fragments are also equivalent, producing many erroneous, useless and even contradictory results.

As a possible solution to the drawbacks of the pure statistical machine translation (weak on reordering; lack of target language fluency), syntactic approaches were proposed that work with traditional statistic models: syntax-based statistical machine translation (Yamada and Knight, 2001); string-based (Galley et al., 2004; Galley et al., 2006); tree-based (Lin, 2004; Liu et al., 2006) forest-based (Mi et al., 2008); forest pruning based systems (Mi et al., 2008). These methods perform better than the non-statistical ones, but require large bilingual corpora.

One other major problem of statistical methods is their difficulty in applicability with agglutinative languages. For example, in Hungarian one noun can have more than 2000 possible forms (combinations of number, case, number or person of grammatical possessors or possessed, etc), thus simply collecting enough statistical data is an enormous task. Lemmatizers could facilitate this task, but because of the complexity of the inflection rules and the high number of exceptions from the rule, efficient Hungarian lemmatizers are not yet available.

3 Proposed method

In order to achieve high precision, our method analyzes all instances of a certain rule, attempting to extract the most frequent, and thus the most suitable transfer rules. During this process, it looks for the most general rule as possible, subcategorizing or exemplifying only when needed. Our method follows a bottom-to-top mechanism, looking to identify not only general translation templates, but also *partial rules*, or frequent subsequences of a certain pattern, mainly targeting inflections and conjugations.

3.1 Resource details

To generate the transfer rules, the proposed method uses an automatically generated *bilingual dictionary* (Varga and Yokoyama, 2007) and a *parsed bilingual corpus*.

There is no known large digital bilingual corpus for Hungarian and Japanese, therefore we needed to improvise with a small, manually created corpus using bilingual language books for Japanese or Hungarian learners that has only about 5000 sentence pairs. 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.

For Hungarian we used *MetaMorph* (Prószéky and Novák, 2005) and for Japanese *Cabocha* (Kudo and Matsumoto, 2000) parsers.

Our method's bilingual corpus requires a specific format. To ensure robustness, we opted for label-bracketed phrase structure rules, since these retain a relatively detailed syntax of the language. For example, the format for the Japanese sentence 夜はオペラに行った [Yoru ha opera ni itta 'Last night I went to the opera'] is *(S (PP (N 夜) (Part は)) (VP (NP (N オペラ)(Part に)) (V 行った))))*.

Both parsers needed minor modifications to accommodate the output. Moreover, the Japanese parser had to be adapted to the *Yamada-grammar*

(Moriyama, 2000), so that one word should represent one concept, similarly with Hungarian.

To be able to generate low-level inflection rules, we extended both Hungarian and Japanese parses with additional, optional information for inflected or conjugated words. The additional information are: (1) information about the inflection or conjugation and (2) the stem of the inflected or conjugated word to be easily identified from the dictionary. For example, the additional information for the Japanese verb 行った [itta; 'went'], becomes *(V<2p> 行った<行く>)*. The bracket after the part-of-speech (POS) information represents the grammatical category of the inflection (2: time; p: past, hence <2p>: past tense), while the bracket after the inflected word represents the stem of the word.

3.2 Transfer rule generation

Our method is composed of two steps: first we generate the language models for each language; next the transfer rules are generated. Both steps are entirely automated.

Step 1: Language model generation

In this step we are looking to build the language model of the two languages. We compute every sentence, saving all partial rules.

We can distinguish four types of rules:

1. *head rules:* rules where the parent is the sentence itself. These are sentence templates, each sentence has exactly one head rule (ex: *S→NP+VP*);
2. *lexical rules:* rules whose children are *lexical categories (POS).* A large percentage of these rules are inflection and conjugation rules, very laborious to manually recreate for agglutinative languages (ex: *VP→V; NP→Adj+N; PP→N+Part*);
3. *terminal rules:* 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 rules:* every rule that is not head, lexical or terminal rule (ex: *NP→Adj+NP*).

We count the frequency of each rule, saving also the sentences from which they were generated.

Head, lexical and regular rules need to be *solved* (transfer rules need to be generated to each of them). We consider a rule to be solved, when there is a correspondence with it in the target language. The transfer rules for the *terminal rules* are the bilingual dictionaries, and thus they are already solved.

Step 2: Transfer rule generation

In this step we build the transfer rules between the two languages. Using a recursive algorithm (solve_rule) we build the transfer rule candidates followed by a noisy rule elimination process (clean_candidates). This is performed twice, once as Hungarian and once as Japanese set as source language.

solve_rule()

For each source head rule (S=s→children(s)), we retrieve all S[] instances (retrieve_instance) in which this rule appears (line 7). If there are S'[] children that are not solved (not_solved), we attempt to solve its children's rules (line 9).

```
1  procedure main();
2     for each s→children(s) in "head rules"
3        solve_rule(s→children(s));
4     clean_candidates();
5
6  procedure solve_rule(s→children(s));
7     S[]=retrieve_instance(children(s)→children(children(s)));
8     if exists S'[]=not_solved (S[])
9        solve_rule(children(s')→children(children(s'))));
10    else
11       TS[]=retrieve_translation(S[]);
12       T[]=identify_match(TS, S[]);
13       generate_candidates(T[], S[]);
14    endif
```

For example, in the case of *S→PP+VP* as a Japanese head rule, an instance where this rule appears is the *PP(夜は)+VP(オペラに行った)* [yoru ha + opera ni itta 'Last night + I went to the opera'] sentence. Since the *PP→N(夜)+Part(は)* and *VP→PP(オペラに)+V(行った)* are not solved yet, the algorithm attempts to solve these first (line 9). When and if these two rules will be solved, the parent rule *(S→PP+VP)* will be reattempted. If all children are solved, transfer rule candidates (generate_candidates) are generated (line 13) using all translations of the rule (retrieve_translation) (line 11).

identify_match()

If a source rule that is investigated has only solved children, we retrieve all instances of the rule, together with their target language correspondences (TS[]). These target language correspondences are whole sentences, we need to identify which parts of these sentences correspond with the source rule. All rules are identified by their children's rules; therefore if the rule in question is a lexical rule, the identification is done using the bilingual dictionary. The stemmed expression is vital in this case. If no information about the stem is available, there is a risk that the word will not be retrieved from the dictionary. If the rule is a head rule, the identification is done using the already solved rules, while with regular rules both resources are needed.

In case of lexical rules, we look up each lexical category's instance (stemmed word, if it is 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 target 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 a transfer rule candidate (Figure 1). For example, to identify the $s_1 \rightarrow s_2+s_3$ lexical rule from the $t_1 \rightarrow t_2+t_3$ sub-tree, the $s_2 \rightarrow w_4$ and $s_3 \rightarrow w_5$ terminal rules are looked up using the dictionary. The process can have multiple scenarios: the words correspond to the same sub-tree (scenario1), or to different sub-tree (scenario2). In the latter case, multiple candidates ($t_2 \rightarrow t_4+t_5$ and $t_3 \rightarrow t_6+t_7$) are saved.

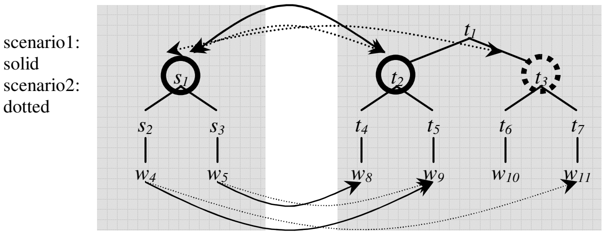

Figure 1: Lexical rule candidate scenarios for $s_1 \rightarrow s_2+s_3$

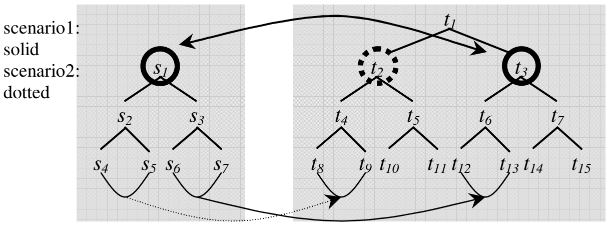

Figure 2: Head and general rule transfer rule candidate scenarios for $s_1 \rightarrow s_2+s_3$

In case of head and regular rules the only difference is the number of children's children. While with lexical rules this was one (one instance for each PoS), for non-lexical rules this is generally at least two (Figure 2). If no correspondences are found, no transfer rule candidate is returned.

instantiate()

In case of lexical rules there are cases when the translations are not registered in the dictionary. In these cases, assuming that the dictionaries are correct, these words have a grammatical, rather than a lexical function. In this case the corresponding lexical category is *instantiated*, being replaced by its instance. Instantiation is performed also when inflection or conjugation information are available.

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

Table 2: Hungarian transfer rule candidates for PP→N+は[ha] (unfit candidates with italic)

For example, if our initial $s_1 \rightarrow s_2(w_1)+s_3(w_2)$ rule's w_2 word did not have any correspondence, the rule becomes $s_1 \rightarrow s_2(w_1)+w_2$. For example, in case of the Japanese *PP→N+Part*, there is no regular rule for a noun plus a particle, therefore the method correctly makes the judgment that the particle needs to be instantiated and new rules are generated for each instance (Table 2).

clean_candidates()

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, definite articles (*a, az*) (English: *the*) also don't have translations in the Japanese rule, but it they have no translation in the dictionary either (there is no corresponding Japanese translation), therefore they were allowed.

The remaining candidates are grouped by their common nodes and are saved with three values:

total nr of candidates; total nr of transfer rule instances; nr of instances for the current rule. Since we do not use any thresholds within our method, these three numbers indicate the confidence level of the transfer rule. For example, for *PP→N+*は [ha] only one transfer rules could be generated: *NP→DET+N<2s>*. The corresponding values are (7, 2, 2).

4 Evaluation

For evaluation, we fragmented our corpus into 5 fragments. First we randomly separated 100 sentences that we used these as our evaluation data. Next we randomly separated 4 training corpora of 100, 500, 1000 and 2000 sentence pairs, to analyze the score differences across various sized corpora. Due to the small size of the available Hungarian-Japanese corpus, performing BLEU score was not adequate, since not enough statistical data was available. Instead, we performed automatic recall evaluation and a manual adequacy evaluation to evaluate our method. We used the rules whose *number of instances for the current rule* is at least 2.

4.1 Recall evaluation

We investigated to what percentage our method's output rules (R_o) manage to cover the training data's phrase structure rules (R_T). We performed a weighted recall evaluation, weighting each rule (r) with its frequency (*frequency*(r)) 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.

$$recall = \frac{\sum_{r \in R_O} frequency(r)}{\sum_{r \in R_T} frequency(r)} \cdot 100 \qquad (1)$$

We analyzed the Japanese coverage, separately evaluating the *head*, *regular* 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 (Table 3). 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.

Rule type	**Training size**			
	100	**500**	**1000**	**2000**
head rules	27.27	40.00	42.67	45.98
regular rules	35.23	41.94	49.65	57.36
lexical rules	47.71	53.12	61.32	64.23

Table 3: Weighted recall evaluation results

4.2 Adequacy evaluation

We automatically simulated a basic rule based machine translation system with 100 Japanese sentences whose rule head has a transfer rule. These sentences were randomly selected from the test data of our bilingual corpus. Our simple machine translation system exhaustively applied all suitable transfer rules, also performing lexical transfer, based on the bilingual dictionary. During this process, all intermediate data (lexical, regular and head rules) was separately saved. As a result, multiple Hungarian translations became available for a single Japanese sentence.

Next, all Hungarian translations, together with the reference retrieved from the bilingual corpus, were manually checked by 3 independent, Hungarian native speakers. We used a 5 to 1 scoring criteria, where 5 is a *perfect*, 1 is a *totally wrong* output sentence. The interpretation of the intermediate scores was left to the judgment of each evaluator. We separately evaluated the *head*, *regular* and *lexical* rules. We performed the same evaluation on four training corpora: 100, 500, 1000 and 2000 samples.

Rule type	**Training size**			
	100	**500**	**1000**	**2000**
head rules	1.42	2.15	2.34	2.67
regular rules	2.11	2.41	2.89	3.22
lexical rules	2.67	3.46	3.97	4.12

Table 4: Adequacy evaluation results

Assuming that a language model would correctly identify the best translation, we considered only the best scoring Hungarian translation for each Japanese source. Lexical items scored best, since understandably the errors on the lower level reflected within the regular 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 regular and head rules (Table 4).

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[watashi]+は[ha] (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, because it was automatically generated using a pivot language (Varga and Yokoyama, 2007). 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 relative recall increase was between 3%-11%, the relative adequacy increase between 4%-13%. This significant increase shows that with an average sized corpus much better results can be achieved.

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 didn't handle grammatical exceptions 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 adequacy, 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.

Cicekli, I., Güvenir, H. A. 2003. Learning Translation Templates from Bilingual Translation Examples, *Recent Advances in Example-Based Machine Translation*, Kluwer Publishers, pp. 255-286.

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.

László, A. 1977: Towards a new Hungarian grammar (Egy új magyar nyelvtan felé), *Magvető Kiadó*, Budapest. *(in Hungarian)*

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.