# Transformation-based correction of rule-based MT

**Jakob Elming**
Center for Computational Modelling of Language
Copenhagen Business School
Denmark

je.id@cbs.dk

### Abstract

We present a pilot study for using transformation-based learning for automatic correction of rule-based machine translation. Correction rules are learned based on a parallel corpus of machine translations from a commercial machine translation system and a human-corrected version of these translations. The correction rules exploit information on word forms and part of speech. The experiment results in a relative increase in translation quality of 4.6% measured using the BLEU metric.

## 1 Introduction

The situation in commercial MT today reveals that most systems are based on rules crafted by language experts. The ideas behind these rule-based systems, however, leave the language experts with bad odds. Not only do they have to give a full description of two languages, but they are also expected to specify how to get from one to the other. The experiences from previous attempts, e.g. the Eurotra project, has shown that it is in fact a very difficult task, which leaves many uncovered areas.

One of the great advantages of rule-based MT as opposed to e.g. statistical MT, is its transparency. Basing the translation on rules makes the output — and thereby the errors — very consistent. If the system makes an error once, it will most likely repeat this type of error in future translations.

This aspect of rule-based MT makes it interesting from a machine learning point of view. The ability to predict recurring errors and their corrections is a simple method for improving an MT system; either by means of a post-processing module, or by correcting the rules of the system.

In the present experiment, transformation-based learning (TBL) is utilized as learning algorithm for extracting rules to correct MT output by means of a post-processing module. The algorithm learns from a parallel corpus of MT output and human-corrected versions of this output. The machine translations are provided by a commercial MT system, PaTrans, which is based on Eurotra. In order to add a level of general linguistic knowledge, the data is annotated with part of speech tags. This method results in a substantial improvement of the translation quality of the MT system.

For this pilot study, we only look at corrections involving substitution.

## 2 Method

The goal of the present experiment is to learn the process of correcting recurring errors made by the MT system. This goal is achieved by first word aligning the machine translation to its human-corrected version. This alignment will to a large part consist of links between the same word forms[1] at the same sentence positions as exemplified by figure 1. Two sentences only differ where corrections have been made.

Secondly, based on a given word alignment, a relevant correction is identified as a recurring link between differences — either in word form or sentence position — in a

[1] 'word form' here includes any token of a text e.g. punctuation.

|  |  |  |
|---|---|---|
| Et | — | Et |
| **tal** | — | **antal** |
| af | — | af |
| andre | — | andre |
| eksperimenter | — | eksperimenter |
| . . . |  | . . . |

Figure 1: An example alignment between machine translation (left) and its human correction (Eng. 'A number of other experiments . . . ').

given context. An example of this is the link between the different word forms *tal* and *antal* in figure 1. If this correlation is frequent enough in a given context throughout the training data — e.g. preceded by *et* and followed by *af* — then it is considered as a potential candidate for a correction rule.

These corrections are located by the machine learning algorithm based on manually pre-determined success criteria.

The following two subsections describe the TBL algorithm and the human corrections it is to learn.

## 2.1 Transformation-based learning

One of the first — and probably the best known — application for TBL was part of speech tagging (Brill, 1995). Since then, the algorithm has been applied to such diverse tasks as text chunking (Ramshaw & Marcus, 1995), dialogue act tagging (Samuel, Carberry, & Vijay-Shanker, 1998), and ellipsis resolution (Hardt, 1998).

TBL operates by always emitting the most error-reducing rule at the current state of training. It then applies this rule to the training data and continues its search for more rules in the new corrected training data, until the error-reduction gets below a pre-defined threshold. This leaves a prioritized list of rules which should be applied in the proscribed order, to give the maximal error-reduction. A consequence of this learning strategy is that exceptions to more general rules can be learned at a later stage, thereby correcting overgeneralizations made by these.

The major reason for choosing TBL as learning algorithm for this experiment, is the nature of the rules learned by the system. TBL has the advantage that its rules are linguistically very informative. Looking much like classical rewrite rules, these rules give a very good, linguistically founded, answer to questions concerning where and why errors occur. This allows us to proceed further than mere automatic correction. Through the rules, we obtain information that lets us analyze the types of error made by the system.

The present experiment uses the $\mu$-TBL tool (Lager, 1999), which is a prolog-based implementation of TBL.

## 2.2 The human corrections

Analyzing the data revealed that almost all corrections made to the MT output can be placed in one of two categories:

1. Substitution
2. Re-ordering

The first category will prototypically contain corrections where words have been replaced by other words. The category, however, also includes the addition of extra words (i.e. the substitution of nothing with something) and the deletion of words (i.e. the substitution of something with nothing).

The substitution type of correction is exemplified in figure 2 by *det* which is deleted, *lige så* which is replaced by *også*, and *ske* which is replaced by *finde sted*.

Learning this type of correction is a task that fits TBL very well. The system has to learn in which contexts a word is changed to another. For example, that the Danish preposition *for* (Eng. 'for') should be changed to the preposition *til* (Eng. 'to') if the previous word is *anvendes* (Eng. 'is used'). This is a task that is very similar to the part of speech tagging that TBL was developed for. However, it is not the POS tags which are changed, but rather the word forms.

The second type of correction deals with words that appear in both the MT output and the corrected text, but at different positions. These re-orderings apply both at

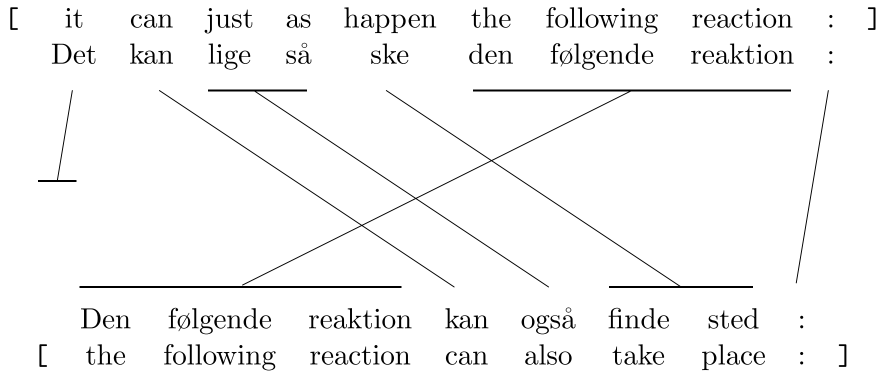
[ it can just as happen the following reaction : ]
Det kan lige så ske den følgende reaktion :
Den følgende reaktion kan også finde sted :
[ the following reaction can also take place : ]

Figure 2: Example of relations between machine translation (on top) and corrected text (literal English glosses in square brackets).

word level and phrase level. This correction type is exemplified in figure 2 by the noun phrase *den følgende reaktion* (Eng. 'the following reaction') which is moved to the front of the sentence.

The pilot study described here will restrict itself to learning the substitution type of correction.

## 3 Data

A parallel corpus of Danish machine translation and a human-corrected version of this is used. The texts have been translated from English to Danish using the rule-based MT system PaTrans (Ørsnes, Music, & Maegaard, 1996); a commercial system building on EUROTRA (Copeland, Durand, Krauwer, & Maegaard, 1991) and specialized to translate patent texts. All the texts are therefore in the domain of (chemical) patents. After being machine translated, a professional, human translator corrected the texts.

The parallel texts is automatically word aligned using the Uplug word alignment tool (Tiedemann, 1999), and the machine translation part is POS tagged by a TBL trained tagger (Brill, 1995). Based on the word alignments, the texts are converted to the format presented in figure 3. This format is chosen in order for it to be compatible with $\mu$-TBL.

The excerpt presented in figure 3 shows the noun phrase, “de nylig opfindelser” (Eng. 'the recent inventions'), which has been corrected to “de foreliggende opfindelser” (Eng. 'the present inventions'). As an example, the fourth, fifth and sixth line of the excerpt state that the 124th word of the text is *nylig* (Eng. 'recent'), it is an adjective, and it corresponds to the word *foreliggende* (Eng. 'present') in the corrected text.

```
wd(123,'de').
tag(123,'PRON_DEMO').
wd('de','de',123).
wd(124,'nylig').
tag(124,'ADJ').
wd('nylig','foreliggende',124).
wd(125,'opfindelser').
tag(125,'N').
wd('opfindelser','opfindelser',
   125).
```

Figure 3: The $\mu$-TBL data format used to train the algorithm.

The corpus material consists of 34 texts comprising some 265,000 words. These data are split randomly into three subsets;

- a training set (26 texts ∼ 12,000 sentences ∼ 220,000 words)
- a validation set for evaluation during development (4 texts ∼ 2,000 sentences ∼ 25,000 words)
- a test set for final evaluation (4 texts ∼ 1,200 sentences ∼ 20,000 words)

# 4 Experiment

In conducting the experiment, two areas had to be clarified. First of all, the settings for the TBL system, and secondly, how to evaluate the performance of the extracted correction rules. The following two subsections deals with these issues.

## 4.1 The learning algorithm

The TBL algorithm learns an ordered sequence of transformation rules, which when applied to the data, should produce the largest decrease in errors. The system is, however, not able to create the rules from scratch. It needs a list of rule templates which it can try to instantiate as concrete rules. The templates have the following form:

```
wd:A>B
      <- wd:C@[-1,-2] &
         tag:D@[1,2]
```

The capital letters are variables which the system can instantiate with words or POS tags, and the numbers indicate positions. So the example template states that word A is replaced by word B if word C is one of the two previous words, and one of the two following words carry the POS tag D.

A total of 70 templates are used in the experiment. They specify different combinations of possible contextual influence on substitution. All templates are based on the 6 nearest words. That is, the contexts used are based on three words to each side. There may also be no influence from the context, leading to rules stating that a given word should always be changed into another.

In addition to the template rule, the system is provided with an accuracy threshold and a score threshold. The accuracy threshold states that every rule should at least be successful in 50% of its corrections. The score threshold states that every rule should at least make three more good corrections than it makes bad corrections. If no rule obeys these thresholds, the algorithm stops learning.

## 4.2 Evaluating the corrections

In addition to training, $\mu$-TBL applies the rules to the held out validation data set and calculated an F-score. This F-score indicates how many of the words in the text are correct before and after applying the rules.

The goal of the experiment is, however, to use machine learning for automatic correction of MT output, and good corrections should in turn lead to improved translation quality. But even though the F-score provided by $\mu$-TBL clearly reflects the quality of translation, since a higher F-score indicates that a text has more in common with the gold standard, it does not take the order of the words into account.

In addition to this, the F-score supplied by $\mu$-TBL is based on the $\mu$-TBL formatted data. This is, however, not an exact replica of the original texts, since text structure has been removed, and a certain margin of error must be expected from the automatic word alignment.

To deal with the measurement of translation quality, one possibility often used in machine translation is the BLEU metric, which has been shown to reflect human judgments of translation quality with high accuracy (Papineni, Roukos, Ward, & Zhu, 2002).

In order to evaluate the rules, they are therefore applied to the original, unformatted[2] texts. This is done in order to get a more precise idea of the effect of the post-processing. The BLEU metric is applied to both the original MT output and the automatically corrected version of the held out test data set. For both texts, the human-corrected version is used as gold standard, making their results comparable.

It should be noted here that rather high BLEU score are expected, since the gold standard is generated from the machine translation. This also means that the BLEU metric constitutes an excellent quality measure in this particular task, since every difference between machine translation and gold standard truly is an error. At least it was found important enough by the human

[2] The texts are not entirely unformatted. In order to comply with the BLEU metric and the extracted rules, the texts have been sentence aligned and tokenized.

| Validation set | | |
|---|---|---|
| | MT output | Correction |
| BLEU score | 72.2 | 73.6 |

Table 1: BLEU scores for MT output and corrected output evaluated on the validation set.

| Test set | | |
|---|---|---|
| | MT output | Correction |
| BLEU score | 59.5 | 63.5 |

Table 2: BLEU scores for MT output and corrected output evaluated on the test set.

| Merged set | | |
|---|---|---|
| | MT output | Correction |
| BLEU score | 64.6 | 67.6 |

Table 3: BLEU scores for MT output and corrected output evaluated on the merged set.

translator to correct. So if the corrections make the machine translation look more like the gold standard, this must mean less errors.

## 5 Results

During the training period of the TBL algorithm, the rules are evaluated on the held out validation set. This is done in order to avoid "training" on the test set, i.e. improving the algorithm to do well when evaluated on this particular test set.

Table 1 shows the effect of applying the correction rules to the validation set. The BLEU score is calculated by comparing the original MT output to the gold standard, and then comparing the corrected version of the MT output to the gold standard. The translation quality increases 1.4 measured on the BLEU scale. This correspond to a relative increase in translation quality of 2.0 %.

Finally, the rules are applied to the unseen test data set, and the BLEU metric is used to calculate the quality of the before and after texts. As shown in table 2, the increase in translation quality is 4.0 measured in BLEU score. This corresponds to a relative increase in translation quality of 6.7%, which constitutes a substantial improvement of the translation quality.

```
wd:der>’, der’
      <- wd:af@[-1,-2,-3] &
         wd:’ID NO’@[1,2,3] &
         tag:’N’@[-1,-2,-3] &
         tag:’EGEN’@[1,2,3]
wd:’ID NO’>’nr.’
      <- tag:’EGEN’@[-1,-2,-3]
wd:’SEQ’>’sekvens-ID’
      <- tag:’EGEN’@[1,2,3]
wd:der>’, der’
      <- tag:’N’@[-1]
```

Figure 4: The top 4 rules extracted by the learning algorithm.

The large difference in increase between the two unseen data sets is probably due to the content of the individual texts that the sets contain. We therefore merged the two unseen data sets to one, in order to get a more covering data set to evaluate the correction rules on.

The results of merging the sets are shown in table 3. The average increase achieved by merging the two unseen data sets is 3.0 measured in BLEU score. This still constitutes a substantial relative increase in translation quality of 4.6%.

## 6 Discussion

A large advantage of using TBL is the transparency of its experience. A closer look at the rules that were extracted by $\mu$-TBL, reveals a lot about their quality. In this section, we will take a closer look at the correction rules extracted by the algorithm.

### 6.1 The extracted rules

The training resulted in 1736 correction rules. The top 4 rules are listed in figure 4 as reference.

The algorithm is, of course, very sensitive to the data which it is trained on. For example, one of the largest texts contain 708 instances of the string “SEQ ID NO”. This sequence does not occur in any of the other 33 texts that make up the corpus. In spite of this, the three first rules are actually based on this string. The first uses it as context to add a comma in front of the relative pronoun *der*, which is correct in Danish. And in the two following rules, the string itself is replaced by its Danish equivalent “sekvens-ID nr.” in two steps.

All in all, the three first rules make valid corrections, but unfortunately they are not general. As a consequence, they do not apply when correcting the validation and test sets. This is also the case for many other rules. When tested on the merged unseen data set, only 350 of the 1736 rules applied. Of these, only 62 applied more than 10 times. The reason for this is most likely the relatively small training data set of only 26 texts.

The fourth rule in figure 4 is on the other hand very general. It makes the same correction as the first rule, but in a more general context (if previous word is a noun). This rule applied 542 times to the merged unseen data set (117 times to the validation set and 425 to the test set).

Now we will take a closer look at the general rules that were extracted. That is to say, the 350 rules that applied to the unseen data.

## 6.2 The general rules

A lot of the rules that apply beyond the training material, seem to clear up collocational restrictions, which is an area that often constitutes a problem to rule-based MT. These rules owe to the fact that the MT system chooses the wrong words in a lot of contexts that require specific words. At other points, the system merely makes the wrong choice when translating a polysemous word.

Figure 5 shows 5 rules exemplifying different aspects of these conditions. The first deals with choosing the correct preposition in a given context. The collocation *used for* should always be translated by *anvendt til*, even though *for* is more equivalent to *for* in Danish. The rule states that *for* should be changed to *til* if one of the previous two words is *anvendt*. This leaves open the possibility of a potential adverb like *ikke* (Eng. ’not’) appearing in between the two words.

```
wd:for>til
      <- wd:anvendt@[-1,-2]
wd:om>ca.
      <- tag:'NUM'@[1,2]
wd:af>ifølge
      <- wd:krav@[1]
wd:h>timer
      <- tag:'NUM'@[-1,-2,-3]
wd:beskrives>'er beskrevet'
      <- tag:'N'@[-1,-2,-3] &
         tag:'N'@[1,2,3]
```

Figure 5: 5 example rules extracted by the learning algorithm.

The rule applied 26 times to the merged data set. Of these 19 resulted in a better translation. In other words, the rule has an accuracy of 73% on the unseen data.

The second rule states that *om* should be changed to *ca.*. This reflects the fact that the polysemous English word *about* can be translated to both *om* and *ca.*, but in the ’approximately’ sense, *om* is not an option in Danish. The rules narrows down the ’approximately’ sense of *about* by stating that the change should take place if one of the following two words is a number.

This rule was also very successful. It applied 115 times to the merged unseen data set, all of which resulted in better translations.

The third rule exemplified in figure 5 specifies that *af* should be changed to *ifølge* if the following word is *krav*. This relies on the fact that the MT system translates the sequence *according to claim* by *af krav*. The professional translators have, however, found *ifølge krav* to be a better translation.

This rule applied 21 times to the merged data set. All of these corrections were good.

The fourth rule concerns the English abbreviation for ’hour’. The system has not translated the letter *h* appearing by itself, either by choice or lack of ability. The rule specifies that this *h* should be changed into *timer* (Eng. 'hours') if one of the three preceding words is a number.

A problem exemplified by this rule, is that the rules would benefit from more generalization. The rule misses the fact that *h* is not necessarily an abbreviation for *hours.* It might as well be an abbreviation for *hour*, which should be translated into *time.*

The rule applies 72 time. All of these lead to improvement in the sense that it is better to have *timer* than *h* if the correct word is *time.* 21 of the corrections, however, yield a wrong inflectional form of TIME (Eng. HOUR). One way of dealing with this problem might be to add lemma form information.

The final rule exemplified in figure 5 is of a more stylistic nature. The English *is described* can be translated into both *beskrives* and *er beskrevet* in Danish. The first choice is passive voice and the other is active. While the MT system has chosen the first, the professional translator have chosen the second to be a better translation.

This rule specifies that *beskrives* should be changed into *er beskrevet* if one of the three previous words is a noun, and one of the follow three is a noun as well. This probably reflects the use in which 'something is described in something'. The rule applies 19 times to the unseen data of which 18 are successful.

Another problem that is covered extensively in the rules is the placement of comma. This was briefly touched upon and exemplified in connection with the relative pronoun *der* in section 6.1. 51 of the 350 general rules specify contexts in which a comma should be added. This is a big concern when translating between English and Danish, since Danish allows comma in a lot of places where English does not.

## 7 Related work

Utilizing the relation between MT output and a human-corrected version of this output is an inviting thought. If it is possible for a computer to learn systematic behavior in the correction process, it will be able to employ these observations to improve quality in future translations. The area is, however, not well examined.

Font Llitjós, Carbonell, & Lavie (2005) have proposed automatic refinement of rules in a rule-based MT system. In their approach, naïve bilingual speakers are used to correct MT output, while the computer records their actions. This leads to very informative data where the informants have to state not only their correction (e.g. that a certain word is replaced by another), but also which word in the context triggered this. Based on the information provided by the informants, the original rules of the MT system are modified or new rules are added.

Their approach is first and foremost aimed towards making MT available for resource poor languages. This, however, makes it less desirable for MT systems that are already in use, since these often have a lot of available resources. An MT system being used by a commercial translation company will not only have a lot of MT output, but also corrected versions of this output. In order to improve on such a system, it would therefore seem a waste to start from scratch.

George & Japkowicz (2005) also uses machine learning techniques to learn corrections for rule-based MT. Focusing on the problem of relative pronoun translation between French and English, they employ different machine learning strategies to detect and correct wrong translations of the relative pronoun. The algorithm is trained on a small corpus of wrongly translated relative pronouns and their correct counterparts. It is also provided with information on part of speech and the semantics of noun phrases.

Based on the experience of the learning algorithm, the are able to detect an incorrectly translated relative pronoun with an accuracy of 83.7%, and 73.1% of their corrections are successful.

## 8 Conclusion

The goal of this pilot study was to use transformation-based learning for automatic correction of rules-based machine translation. By learning context-dependent substitution rules based on word forms and part of speech tags, a substantial increase in translation quality was achieved. Measured on the BLEU metric, the average increase was 3.0 from 64.6 to 67.6, which constitutes a relative increase in translation quality of 4.6%. This shows that it is possible by relatively simple means to locate recurring errors in rule-based MT, correct them, and thereby improve the translation quality.

## 9 Future work

The promising result of this pilot study motivates further work on the project.

The most important next step will be to include learning the re-ordering corrections as well. This means adding more levels of representation, e.g. a phrase structure level. Since one of the great problems of MT is word re-ordering, it seems reasonable to assume that learning this type of correction would improve translation quality at least as much as learning the substitution rules.

Furthermore, corrections sometimes incorporate both types of correction, e.g. a word is changed into another only if it is at the same time moved to another position. We saw an example of this in figure 2. Here, the noun phrase can only be moved to the initial position, if this position is empty. This is achieved by deleting the dummy subject *det*.

Additional improvement to the system would probably also involve leaving the $\mu$-TBL platform. This would bring about more freedom to handle additional representation levels and include other parameters for the system to evaluate its experience. In more specific terms, it would be interesting to rank the candidate rules based on the increase in BLEU score they bring about. This would bring word order into the picture in addition to the number of correct words.

## References

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics.*

Copeland, C., Durand, J., Krauwer, S., & Maegaard, B. (1991). The eurotra linguistic specifications. In *Studies in machine translation and natural language processing* (Vol. 1). Luxembourg: Office for Official Publications of the Commission of the European Community.

Font Llitjós, A., Carbonell, J., & Lavie, A. (2005). A framework for interactive and automatic refinement of transfer-based machine translation. In *Proceedings of the eamt 10th annual conference.* Budapest, Hungary.

George, C., & Japkowicz, N. (2005). Automatic correction of french to english relative pronoun translations using natural language processing and machine learning techniques. In *Computational linguistics in the north east (cline'05).* Ottawa.

Hardt, D. (1998). Improving ellipsis resolution with transformation-based learning. In *Aaai fall symposium.* .

Lager, T. (1999). The $\mu$-tbl system: Logic programming tools for transformation-based learning. In *Proceedings of the third international workshop on computational natural language learning.* Bergen.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Philadelphia.

Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. In *Proceedings of the acl third workshop on very large corpora* (pp. 82–94). .

Samuel, K., Carberry, S., & Vijay-Shanker, K. (1998). Dialogue act tagging with transformation-based learning. In *Proceedings of coling/acl'98.* .

Tiedemann, J. (1999). Uplug - a modular corpus tool for parallel corpora. In L. Borin (Ed.), *Parallel corpora, parallel worlds.* Amsterdam: Rodopi.

Ørsnes, B., Music, B., & Maegaard, B. (1996). Patrans – a patent translation system. In *Proceedings of coling* (pp. 1115 – 1118). Copenhagen.