WICKET – Word-aligned Incremental Corpus-based Korean-English Translation

Werner Winiwarter

University of Vienna, Department of Scientific Computing
Universitätsstraße 5, A-1010 Wien
werner.winiwarter@univie.ac.at

Abstract. In this paper we present a Korean-English machine translation system. In our approach we use a transfer-based machine translation architecture, however, we learn all the transfer rules automatically from translation examples by using structural alignment between the parse trees. We provide the user with a comfortable Web interface to display detailed information about lexical, syntactic, and translation knowledge. This makes our system also a very useful tool for computer-assisted language learning. The linguistic knowledge, including lexicons and grammars, is learnt automatically from a Korean-English treebank. The only required additional input for rule acquisition are word alignments. For this task we offer a user-friendly Web interface with simple drag-and-drop operations. The system has been implemented in Amzi! Prolog, using the Amzi! Logic Server CGI Interface to develop the Web application.

Introduction

Despite the huge amount of effort invested in the development of machine translation systems, the achieved translation quality is most often still disappointing. One major reason is the missing ability to learn from translation errors through incremental updates of the rule base.

In our research we use the bilingual data from the Korean-English Treebank Annotations by the Institute for Research in Cognitive Science, University of Pennsylvania [Palmer et al. 2002] as training material. The treebank consists of 5083 Korean-English sentence pairs, which have been manually annotated, including syntactic constituent bracketing and part-of-speech tagging. We use the parallel treebank to automatically learn lexicons and grammars for both source and target language. With the assistance of a user-friendly Web interface we add word alignments to the treebank by using simple drag-and drop operations. This enriched treebank is then used to learn transfer rules through structural matching between the syntactic representations of the examples in the source and target language.

Our current research work originates from the JETCAT project (Japanese-English Translation using Corpus-based Acquisition of Transfer rules, [Winiwarter 2008]) in which we had developed a translation system from Japanese into English. One main research goal of our current activities was to show that the our approach is truly generic, i.e. the acquisition, representation, and application of transfer knowledge is language-independent. This means that the research challenge was to show that the Japanese-English machine translation system could be adapted to Korean-English translation with minimal effort.

For the implementation of our machine translation system we have chosen Amzi! Prolog because it provides an expressive declarative programming language within the Eclipse Platform.

It offers powerful unification operations required for the efficient application of transfer rules and full Unicode support so that Korean characters can be used as textual elements in the Prolog source code. Amzi! Prolog comes with several APIs, in particular the Amzi! Logic Server CGI Interface, which we used to develop our Web interface.

Related Work

The research on machine translation has a long tradition [Hutchins 2001]. The state of the art in machine translation is that there are quite good solutions for narrow application domains with a limited vocabulary and concept space. However, it is the general opinion that fully automatic high quality translation without any limitations on the subject and without any human intervention is far beyond the scope of today's machine translation technology and there is serious doubt that it will be ever possible in the future [Hutchins 2003].

It is very disappointing to notice that the translation quality has not much improved in the last few years [Somers 2003]. One main obstacle on the way to achieving better translation quality is seen in the fact that most of the current machine translation systems are not able to learn from their mistakes [Hutchins 2004]. Most of the translation systems consist of large static rule bases with limited coverage, which have been compiled manually with huge intellectual effort. All the valuable effort spent by users on post-editing is usually lost for future translations.

As a solution to this knowledge acquisition bottleneck, *corpus-based machine translation* tries to learn the transfer knowledge automatically on the basis of large bilingual corpora for the language pair [Carl 1999]. *Statistical machine translation* [Brown 1990], in its pure form, uses no additional linguistic knowledge to train both a statistical translation and target language model. The two models are used to assign probabilities to translation candidates and then to choose the candidate with the maximum score. For the first few years the translation model was built only at the word level. Several extensions towards phrase-based translation [Koehn/Och/Marcu 2003] and syntax-based translation [Yamada 2002] have been proposed. Although some improvements in the translation quality could be achieved, statistical machine translation has still one main disadvantage in common with rule-based translation, i.e. an incremental adaptation of the statistical model by the user is usually impossible.

The most prominent approach for the translation of Japanese and Korean has been *example-based machine translation* [Hutchins 2005]. It uses a parallel corpus to create a database of translation examples for source language fragments. The different approaches vary in how they represent these fragments [Carl/Way 2003]: as surface strings, structured representations, generalized templates with variables, etc. However, most of the representations of translation examples used in example-based systems of reasonable size have to be manually crafted or at least reviewed for correctness to achieve sufficient accuracy [Richardson et al. 2001].

System Architecture

The system architecture of WICKET is displayed in Fig. 1. The users work with their Web browsers, which send CGI calls to the Web server and receive dynamically generated Web pages in return. At the Web server the CGI interface communicates with a C program with extended predicates for Prolog and a Prolog program with a library of CGI support predicates.

The middle part of Fig. 1 shows the translation of a Korean sentence. We first perform the tagging of the sentence by accessing the Korean lexicon to produce a Korean token list.

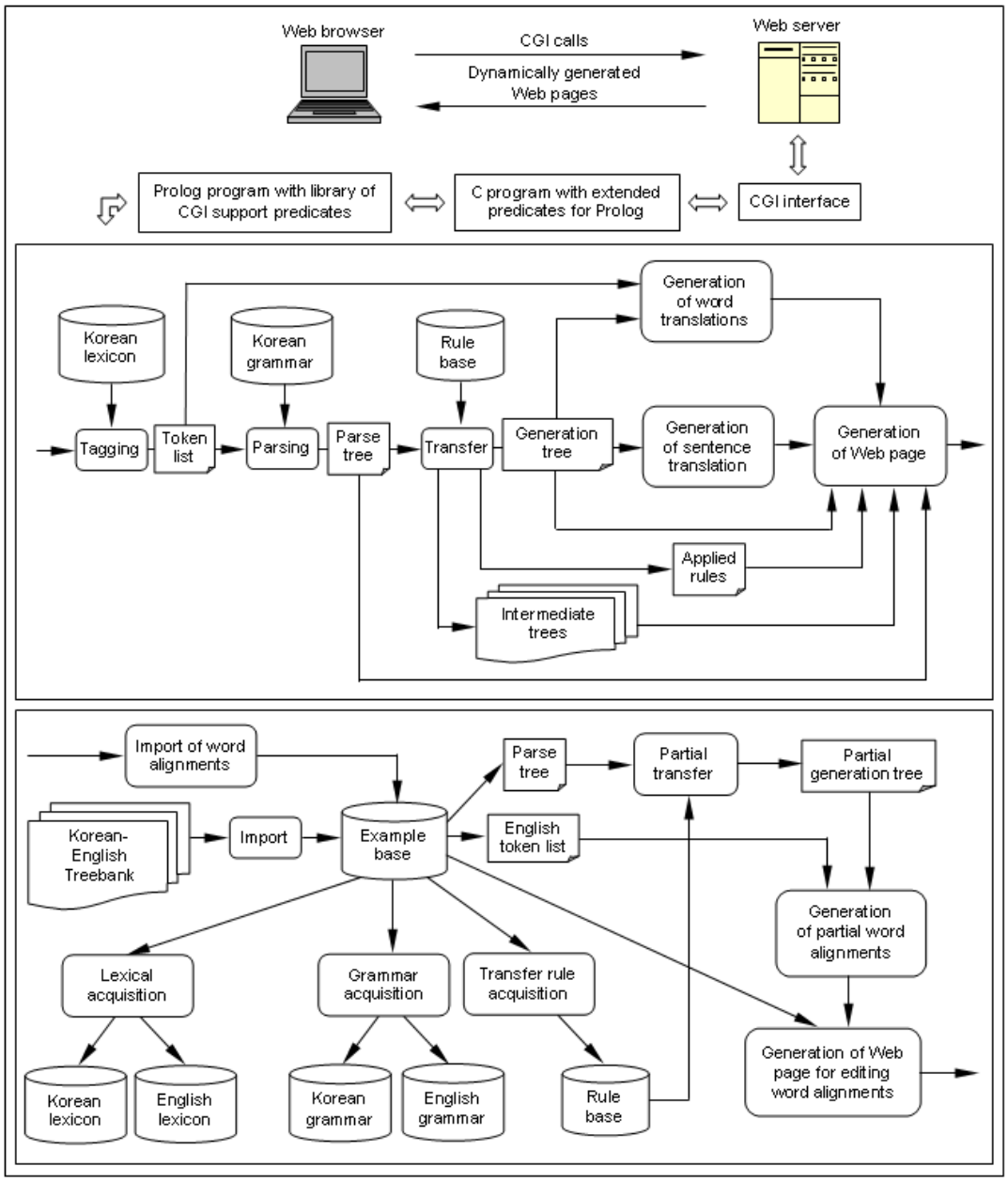

Fig. 1: System architecture

The next step is the parsing of the sentence by applying the Korean grammar rules. During the transfer the Korean parse tree is then transformed into a corresponding English tree, the generation tree, through the application of the transfer rules in the rule base. The final task is the generation of the surface representation of the sentence translation as character string by flattening the structured representation.

In addition to the sentence translation, we also produce context-specific word translations and store the sequence of all applied rules and intermediate trees to send all translation details back to the user.

The acquisition of new linguistic knowledge is depicted in the lower part of Fig. 1. We import the treebank files into the example base to learn the lexicons and grammars for source and target language. For the acquisition of the transfer rules we also require word alignments, which are not provided by the original treebank. We offer a user-friendly Web interface to import word alignments by using simple drag-and-drop operations (see Fig. 2).

To facilitate this task, we suggest candidates for word alignments wherever this is possible. For this purpose we first perform a transfer with the existing transfer rules to produce a partial generation tree. The successfully translated elements are collected as a list and mapped to the elements in the English token list to compute the candidates for word alignments.

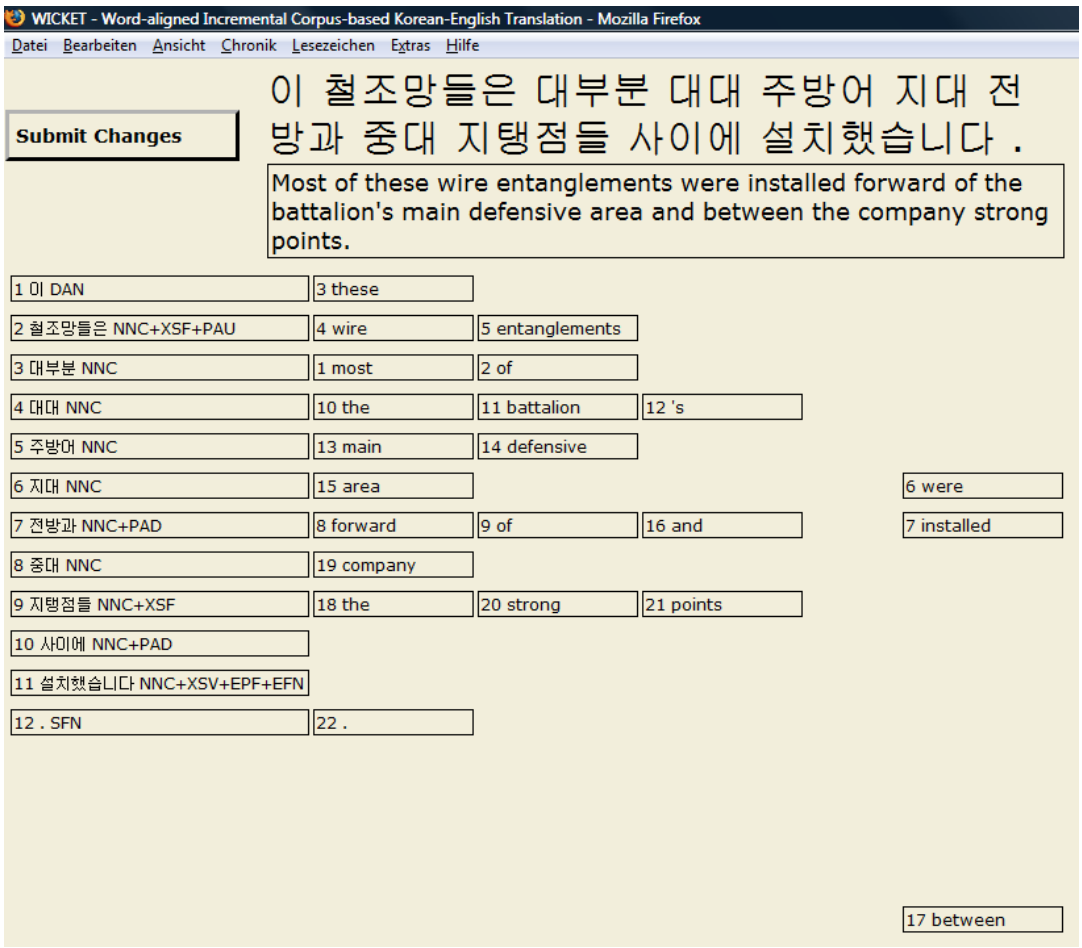

Fig. 2: Screenshot of Web interface for the import of word alignments

Lexical Knowledge

To access the lexical data for a Korean sentence the user has simply to move the mouse over the individual words, which results in the display of pop-up windows indicating the Roman transcription of the Hangul script, the context-specific word translation, and the part-of-speech tag. For inflected lexical forms we also indicate this information for the base form and its inflections (see Fig. 3).

Fig. 3: Screenshot of lexical knowledge

The lexical acquisition module creates a lexicon entry for each new Korean word. For inflected word forms, the base form and its inflections are stored as additional entries. If a word can be used with several different part-of-speech tags, we store one default tag in the lexicon and cover other word meanings by learning word sense disambiguation rules based on the local context, which are also stored in the lexicon. During lexical analysis each word is first tagged with the default part-of-speech tag, which may then be corrected by applying word sense disambiguation rules to consider additional word senses.

The same way we store new English words in the English lexicon. Ambiguous words are again covered with word sense disambiguation rules. The English lexicon is only used for learning new transfer rules from examples for which only surface sentences are available as input.

Syntactic Knowledge

The parse tree for a Korean sentence can be displayed as menu tree with tool tips for all constituents; subtrees can be freely expanded and collapsed (see Fig. 4).

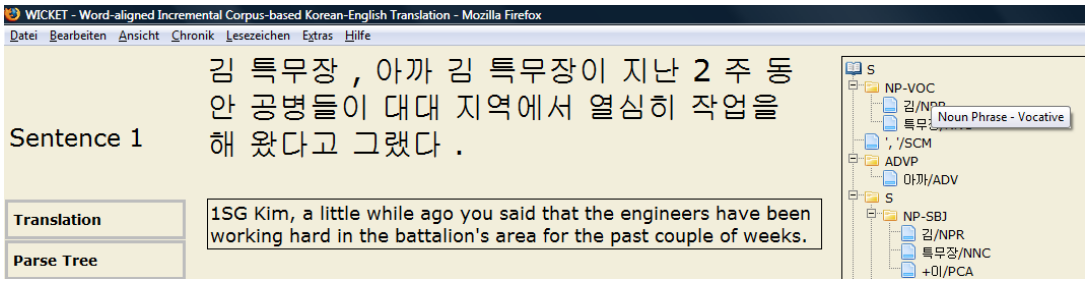

Fig. 4: Screenshot of syntactic knowledge

We model a Korean sentence as a list of *constituents.* A *simple constituent* represents a word with its part-of-speech tag and position index in the token list as `index/word/tag`. We use separate constituents for the base form and the inflections of an inflected word form, the inflections are indicated as `'+ '/inflection/tag`. A *complex constituent* models a phrase as `[category | argument]` where the *argument* is the list of subconstituents.

During grammar acquisition we learn the grammar rules automatically from token lists and parse trees. To parse a Korean sentence, we apply the grammar rules in a bottom-up approach. We first collect all rule candidates that can be applied to the current configuration. Then we choose a rule depending on the number of simple and complex constituents in the condition part. We apply the rule and start the next iteration until no new rule can be applied.

English sentences are represented in the same way. We also learn the English grammar rules automatically from token lists and parse trees. However, we only use the English grammar for learning new transfer rules from examples for which no treebank is available as input.

Translation Knowledge

The user can display the generation tree as well as the sequence of transfer rules that were applied to the Korean parse tree to produce the translation. In addition, it is possible to display all the individual transfer steps, i.e. the intermediate trees before and after applying each transfer rule. The constituents affected by the rule are highlighted by color in the trees (see Fig. 5). The user can just move the mouse over the individual rules in the rule table to obtain an animated view how the Korean parse tree gradually changes into a fully translated English tree.

The rule base is created automatically by using structural matching between parse trees of translation examples from the word-aligned treebank. The acquisition module traverses the Korean and English parse tree for a translation example and derives new transfer rules. The search for new rules starts at the sentence level by recursively mapping the individual subconstituents of the Korean sentence. Before adding new rules we check for side effects on the correct translations for the example base; if necessary, we increase the specificity of the rules.

We distinguish between three rule types: *word transfer rules* translate individual words, *phrase transfer rules* the argument of complex constituents, and *constituent transfer rules* the category and argument of complex constituents.

The acquisition procedure is fully generic, i.e. it uses no linguistic knowledge to guide the learning process. The acquisition is performed only based on the structure of the two trees and the position information from the word alignments. For example, to learn the first rule displayed in Fig. 5, we first search the Korean tree with the following result for the condition part:

[['NP'/'SBJ' | X1], ['VP', ['VP', ['LV', 하/'VV', 었/'EPF', 는가/'EFN'] | X2] | X5], '? '/'SFN']

We also store a record that indicates for the three *variables for unification* X1, X2, and X5 the categories, arguments, and corresponding positions in the English token list. For example, X1 represents a subject, X2 an object, and X5 an adverbial noun phrase and an adverb phrase. After retrieving the translations for the required elements in the condition part (“have done?”), we map the record for X1, X2, and X5 with the remaining elements in the English tree that were collected during the traversal. In most cases we have a direct mapping, as for X1 and X2, otherwise we have to split the variable, as for X5, by binding it with the structure ['ADVP' | X4] | X3]. This way, we can deal with any complex situation for mapping the elements of the two trees.

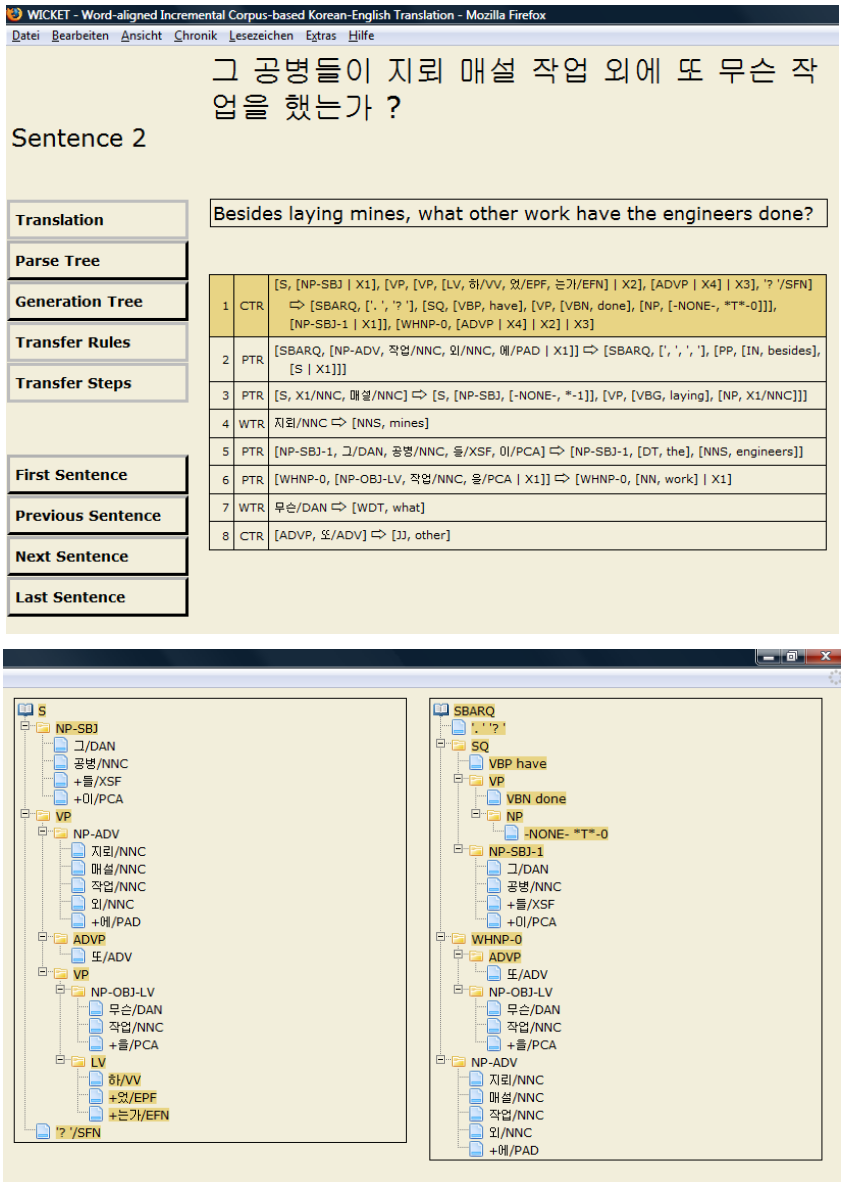

Fig. 5: Screenshot of transfer step

The transfer module traverses the Korean parse tree top-down and searches the rule base for transfer rules that can be applied. We first search for constituent transfer rules before we perform a transfer of the argument. At the argument level we first try to find suitable phrase transfer rules. We collect all rule candidates that satisfy the condition part and then choose the rule with the most specific condition part. If no more rules can be applied, each subconstituent in the argument is examined separately. The latter involves the application of word transfer rules for simple constituents, whereas the procedure is repeated recursively for complex constituents.

Conclusion

In this paper we have presented a Korean-English machine translation system. WICKET learns the transfer rules automatically from a word-aligned treebank. It also displays detailed information about lexical, syntactic, and translation knowledge and offers a Web interface to add word alignments. We have finished the implementation of the system including a first local prototype configuration of the Web server to demonstrate the feasibility of the approach.

Future work will focus on extending the coverage of the system so that we can process the complete treebank and perform a thorough evaluation of the translation quality using tenfold cross-validation. We also plan to make our system available to students of Korean studies at the University of Vienna in order to receive valuable feedback from practical use.

Acknowledgement

This research work has been carried out as part of the bilateral Korean-Austrian pilot project "Interoperability of Ontologies" KR 06/2008 with financial support from the Austrian Federal Ministry of Science and Research.

References

P. Brown. A statistical approach to machine translation. Computational Linguistics, Vol. 16, No. 2, 1990.

M. Carl. Toward a model of competence for corpus-based machine translation. O. Streiter, M. Carl, and J. Haller (eds). Hybrid Approaches to Machine Translation, ser. IAI Working Papers. IAI, Vol. 36, 1999.

M. Carl and A. Way (eds). Recent Advances in Example-Based Machine Translation. Dordrecht: Kluwer, 2003.

J. Hutchins. Machine translation over 50 years. Histoire epistémologie langage, Vol. 23, No. 1, 2001.

J. Hutchins. Has machine translation improved? Some historical comparisons. Proc. of the 9th MT Summit, 2003.

J. Hutchins. Machine translation and computer-based translation tools: What's available and how it's used. J. M. Bravo (ed). A New Spectrum of Translation Studies. Valladolid: University of Valladolid, 2004.

J. Hutchins. Towards a definition of example-based machine translation. Proc. of the 2nd Workshop on Example-Based Machine Translation at MT Summit X, 2005.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. Proc. of the 2003 Conf. of the North American Chapter of the ACL on Human Language Technology, 2003.

M. Palmer et al. Korean English Treebank Annotations. Philadelphia: Linguistic Data Consortium, 2002.

S. Richardson et al. Overcoming the customization bottleneck using example-based MT. Proc. of the ACL Workshop on Data-driven Machine Translation, 2001.

H. Somers (ed). Computers and Translation: A Translator's Guide. Amsterdam: John Benjamins, 2003.

W. Winiwarter. Learning transfer rules for machine translation from parallel corpora. Journal of Digital Information Management, Vol. 6, No. 4, 2008.

K. Yamada. A Syntax-Based Statistical Machine Translation Model. Ph.D. thesis, University of Southern California, 2002.