WETCAT – Web-Enabled Translation Using Corpus-Based Acquisition of Transfer Rules

Werner Winiwarter
Department of Scientific Computing, University of Vienna
werner.winiwarter@univie.ac.at

Abstract

In this paper we present a Web interface to a Japanese-English rule-based machine translation system. One main feature of our translation system is that the transfer rules have not been designed by hand but are learnt automatically from a parallel corpus. The user can customize the rule base by simply correcting translation results. In addition, it is possible to display token lists, parse trees, and transfer rules, which makes our system also a very useful tool for language students. The system has been implemented in Amzi! Prolog, using the Amzi! Logic Server CGI Interface to develop the Web application.

1. Introduction

Studying online documents in a foreign language, readily available through the Web, is an excellent way to improve language skills because new words, phrases, or grammatical constructions are encountered in their natural context. However, for Japanese this approach to language acquisition is hindered by:

1) the complex writing system comprising three different scripts with several thousand characters,
2) the lack of spaces or other visual indicators for word boundaries,
3) the high degree of ambiguity in Japanese grammar, e.g. no articles to indicate gender or definiteness, no declension to indicate number or case, etc.,
4) the tendency to omit any information that can be inferred implicitly, e.g. speaker and addressee in dialogs.

Fortunately, there are many useful tools available to help the user with the difficult task of interpreting a Japanese text. There exist several free Web-based bilingual dictionaries, e.g. the Japanese-English dictionary server WWWJDIC (jp.msmobiles.com) or the Japanese-German dictionary Wadoku Jiten (wadoku.de). Some text editors also provide Japanese-English dictionaries, e.g. MS Office Proofing Tools. Finally, there are Web sites that offer pop-up hints with lexical data (e.g. www.popjisyo.com). Open problems with most of these tools are the correct segmentation into words and the lookup of conjugated words.

Another more direct approach would be to use Web-based machine translation services. Unfortunately, the quality of those free online systems is still quite bad for Japanese.

This situation was our motivation to develop a Web-based high quality machine translation system. In our approach, called WETCAT (Web-Enabled Translation using Corpus-based Acquisition of Transfer rules), we learn all transfer rules automatically by using structural matching between the parse trees of translation examples. The work is based on a previous project on Japanese-German translation [1]. As training material we use the JENAAD corpus [2] comprising 150,000 Japanese-English sentence pairs taken from news articles.

The system was implemented in Amzi! Prolog, which offers an expressive declarative programming language within the Eclipse Platform, powerful unification operations for the efficient application of the transfer rules, and full Unicode support for Japanese characters. Finally, Amzi! Prolog includes several APIs, in particular the Amzi! Logic Server CGI Interface, which we used to develop our Web application. Through the Web interface, the user cannot only translate Japanese sentences but also inspect lexical, syntactic, and transfer information. An important feature is the customization of the rule base by simply post-editing and resubmitting translation results.

The rest of the paper is organized as follows. In Sect. 2 we first provide a brief discussion of related work on machine translation before we give an overview of the system architecture in Sect. 3. The

1-4244-0674-9/06/$20.00

remaining sections of the paper describe the implementation of the individual tasks in more detail.

2. Related work

The state of the art in machine translation is that there are quite good solutions for narrow application domains whereas fully automatic general-purpose high quality translation is still impossible [3]. It is quite disappointing that despite the large effort invested in the development of translation systems, the translation quality has not improved much in the last few years [4].

One main problem of traditional *rule-based machine translation* systems is their static nature, i.e. the limited coverage of the handcrafted rule base cannot be adjusted towards a user's preferences. The most common approach to solve this situation is to learn the translation knowledge based on large parallel corpora.

Statistical machine translation approaches try to learn a translation model and a language model for the target language from a parallel corpus to calculate the most probable translation. Early systems learnt the translation model only at the word level [5] and were therefore only successful for similar language pairs. Recently, several systems have been developed that incorporate extensions towards phrase-based translation [6] and syntax-based translation [7].

Example-based machine translation lies between the two extremes of pure rule-based translation and statistical translation [8]. It also uses a large bilingual corpus to create a database of translation examples. By using a hybrid configuration of different techniques it matches fragments from the user input against source language fragments or pre-compiled representations in the database to retrieve and combine equivalent target language fragments to build the translation [9]. The most common drawback of large-scale example-based machine translation systems is that again some manual effort is required to construct or at least verify the representations of translation examples in the database [10].

3. System architecture

The WETCAT system architecture is depicted in Fig. 1. The user's Web browser sends CGI calls to the Web server, which calls the CGI application to return dynamically generated HTML documents. The CGI application consists of a C program responsible for starting the Amzi! Logic Server and loading the Prolog CGI script. All user input and CGI variables are asserted as facts to the Prolog logicbase before calling the Prolog part of the CGI Amzi! interface. This Prolog wrapper performs the necessary CGI bookkeeping functions and calls predicates defined in the Prolog script implementing the machine translation subsystem.

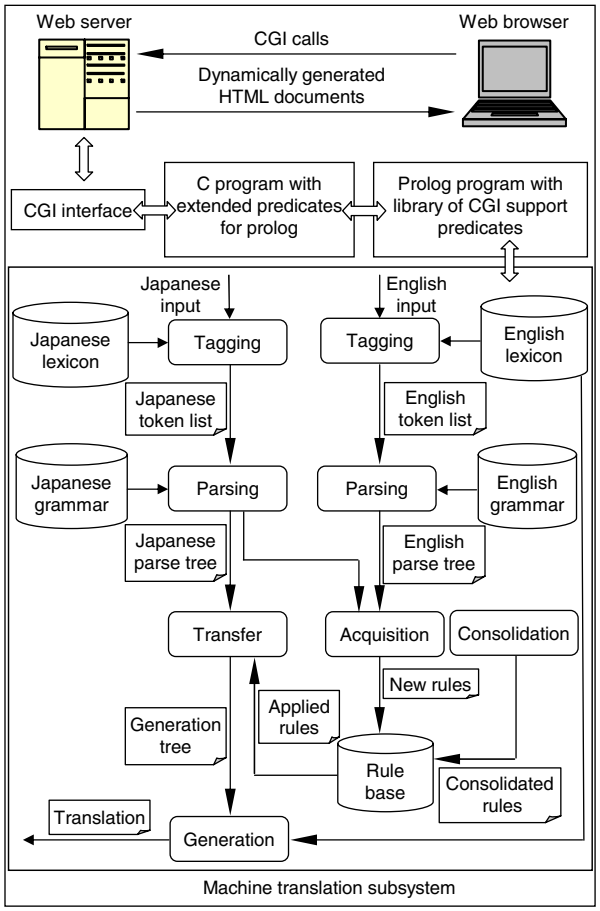

Figure 1. System architecture

The three main tasks for the machine translation subsystem are the translation of Japanese input, the acquisition of new transfer rules, and the consolidation of the rule base. Additional required subtasks are the tagging and parsing of Japanese and English input, the transfer from Japanese to English, and the generation of the English output. All intermediate results can be inspected by the user via the Web interface. The following sections offer a more detailed description of the individual system components.

4. Source language analysis

The first task to be performed is the linguistic analysis of the Japanese input. The user can input a Japanese sentence into the Web interface (see Fig. 2) or use a Visual Basic macro to access WETCAT directly

from MS Word. In the latter case, the user just clicks anywhere in a Japanese document and invokes the macro, which extracts the surrounding sentence and calls the Web server via the GET method by adding the sentence as query string. The Web server responds by returning the Web form with the Japanese input.

Figure 2. Screenshot of Web interface

By clicking on the "Japanese Token List" button the user can retrieve the correct segmentation into word tokens annotated with lexical information. The Japanese sentence is sent to the Web server via the POST method, which returns a list of morphemes with pronunciation, base form, part-of-speech, conjugation type, and conjugation form. For the latter three we use three letter acronyms. Figure 3 shows part of the token list for the sentence in Fig. 2. The output is produced by the *tagging module*, which accesses the Japanese lexicon. The lexicon was compiled automatically by applying ChaSen [11] to the JENAAD corpus.

The user can also view the parse tree for a Japanese sentence by clicking on the "Japanese Parse Tree" button. The *parsing module* computes the parse tree with the assistance of the Definite Clause Grammar preprocessor of Amzi! Prolog by applying the Japanese grammar to the token list. A sentence is modeled as a list of constituents.

日本	ニッポン	日本	prc	-	-
に	ニ	に	xpg	-	-
ダイナミック	ダイナミック	ダイナミック	naj	-	-
な	ナ	だ	xve	ida	ucf
市場	シジョウ	市場	nge	-	-
経済	ケイザイ	経済	nge	-	-
を	ヲ	を	xpg	-	-
確立	カクリツ	確立	nsh	-	-
する	スル	する	vin	ssu	baf
に	ニ	に	xpg	-	-
は	ハ	は	xpo	-	-
、	、	、	syc	-	-
官僚	カンリョウ	官僚	nge	-	-

Figure 3. Screenshot of Japanese token list

A *constituent* is defined as a compound term of arity 1 with the *constituent category* as principal functor. For a *simple constituent*, the argument can be a syntactic feature (atom) or a word with its part-of-speech (atom/atom); for a *complex constituent* it is a phrase (list of subconstituents). A part of the parse tree for the sentence in Fig. 2 can be seen in Fig. 4.

5. Translation

A Japanese sentence is translated by clicking on the "Translation" button. This task is divided in two subtasks: the application of the transfer rules to the Japanese parse tree to produce the generation tree and the generation of the surface string. The transfer rules are stored as Prolog facts in the rule base. We model all translation problems with three generic types of transfer rules. A *word transfer rule* translates the argument of a simple constituent whereas a *constituent transfer rule* changes both the category and the argument of a complex constituent.

Finally, a *phrase transfer rule* allows to define elaborate conditions and substitutions on complex constituents. In the following, we provide three illustrative examples of transfer rules, for a more detailed treatment of our transfer rule formalism we refer to the "ATR presentation" slides at www.amzi.com/customers/winiwarter_xlate.htm:

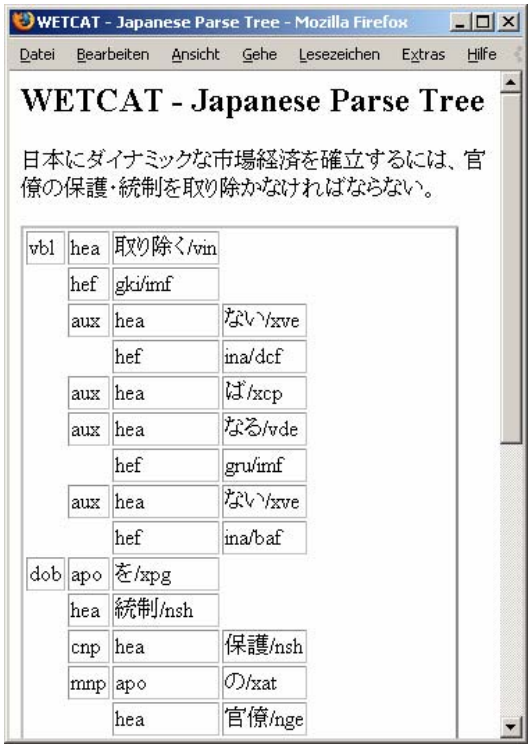

Figure 4. Screenshot of Japanese parse tree

1) wtr(日本/prc, 'Japan'/nnp).
2) ctr(mnp, maj, 官僚/nge,
   [apo(の/xat), hea(官僚/nge)],
   [hea(bureaucratic/jj)]).
3) ptr(cl, 乗り出す/vin,
   [aob([apo(に/xpg), hea(過程/nge) | X])],
   [aob([apo(on/in), hea(process/nn), det(ind) | X])]).

Rule 1 is the default translation of 日本 (NIHON) as 'Japan', Rule 2 changes the modifying noun phrase 官僚の (KANRYOU NO) into the modifying adjective phrase bureaucratic, and Rule 3 states that for a clause with head verb 乗り出す (NORIDASU), the adpositional object X過程に (X KATEI NI) has to be replaced by the adpositional object on a X process.

The third argument of a constituent transfer rule and the second argument of a phrase transfer rule serve as index for the fast retrieval of matching facts to speed up the unifiability test for the condition part of the transfer rules during translation. Both constituent and phrase transfer rules may contain shared variables for unification, as shown in Rule 3. This makes it possible to translate only certain parts of the input and leave the rest unchanged.

One important requirement for the efficient and robust implementation of the transfer module is that the condition in the third argument of a phrase transfer rule has to be understood as subset condition. For example, in Rule 3 it is necessary that the clause contains an adpositional object at an arbitrary position with the adposition に and the head noun 過程, both again at arbitrary positions. All other elements of the clause and of the adpositional object are appended to the translated required elements.

The *transfer module* traverses the Japanese parse tree top-down and searches for transfer rules that can be applied. At the top level we first try to find suitable phrase transfer rules. To apply a phrase transfer rule, we collect all rule candidates that satisfy the condition part and then rate each rule and choose the one with the highest score. If there are no more rule candidates left, we examine each constituent in the sentence individually. We first search for constituent transfer rules before we perform a transfer of the argument. The latter involves the application of word transfer rules for simple constituents, whereas the top-level procedure is repeated recursively for complex constituents.

The resulting generation tree can be inspected through the "Generation Tree" button. In addition, the user can click on "Applied Rules" to receive a list of the rules used by the transfer module in the correct order of their application.

As last step of a translation, the *generation module* produces the surface form of the English sentence as character string. We traverse the generation tree top-down and transform the argument of each complex constituent into a list of surface strings. The list is computed recursively from the subconstituents as a nested list and flattened afterwards.

6. Target language analysis

The tagging and parsing of English sentences is a necessary preprocessing step for the acquisition of new transfer rules from sentence pairs. The English lexicon used by the *tagging module* was built automatically through the application of the MontyTagger [12] to the JENAAD corpus.

The English *parsing module* works on the basis of the English grammar whose rules are again written in Definite Clause Grammar syntax. To facilitate the structural matching during acquisition we tried to align the use of constituent categories in the English grammar as best as possible with corresponding Japanese categories.

The user can view the output of the English tagging and parsing module by clicking on "English Token List" and "English Parse Tree".

7. Acquisition

The WETCAT rule base is built by applying the *acquisition module* to the JENAAD corpus. It traverses Japanese and English parse trees and derives new transfer rules. The search for new rules starts at the sentence level by recursively mapping the individual subconstituents of the Japanese sentence. For each Japanese constituent category there is a specific rule to map a subconstituent of this type (possibly together with other subconstituents) to derive a transfer rule. If a rule could be found, then all subconstituents that are covered by the rule are removed from the Japanese and English input. Each rule is added to the rule base if it is not included yet. The default mapping of a complex subconstituent is to find a corresponding subconstituent in the English input and to continue the matching procedure recursively for their two arguments.

The rules produced by the acquisition module are very specific since they consider context-dependent translation dependencies in full detail to avoid any conflict with other rules in the rule base. This guarantees correct translations but leads to a huge number of complex rules, which badly affects the coverage for new data. Therefore, the *consolidation module* generalizes rules by relaxing their condition part as long as this does not affect any existing rule. The most commonly performed transformations are: to simplify a phrase transfer rule or to replace it with a word transfer rule, to use generalized categories as first argument of phrase transfer rules, and to split a phrase transfer rule in two simpler rules.

For a better understanding of the acquisition task, the user can click on the "New Rules" button, which sends the Japanese and English sentence to the Web server. WETCAT pretends to have an empty rule base and returns a list of rules that would be learnt from this sentence pair in the correct order of derivation. In addition, the user can make use of the "Consolidated Rules" button to look at the generalizing transformations that would be performed for this sentence pair, the result is displayed to the user as a list of "old rule $\Rightarrow$ new rule(s)" pairs.

Finally, one prominent feature of WETCAT is the ability to customize translation results by post-editing the target sentence und updating the rule base via the "Update Rule Base" button. Before this, users can use the "New Rules" and "Consolidated Rules" buttons to verify the effects of their corrections on the acquisition process. As soon as a user commits a change with the "Update Rule Base" button, the sentence will be always translated that way in the future. We keep copies of the default rule base derived from JENAAD for the individual users so that each user can have his own customized WETCAT version.

8. Conclusion

In this paper we have presented a Web-enabled Japanese-English machine translation system based on the automatic acquisition of transfer rules from a parallel corpus. We have finished the implementation of the system for a subset of the JENAAD corpus including a first local prototype configuration of the Web server to demonstrate the feasibility of our approach. Future work will focus on extending the coverage of the system to the complete corpus and on performing a quantitative evaluation. We also plan to make a demo version of WETCAT publicly available in the near future.

References

[1] W. Winiwarter, "Incremental learning of transfer rules for customized machine translation," *Applications of Declarative Programming and Knowledge Management*, U. Seipel et al., eds., Lecture Notes in Artificial Intelligence, vol. 3392, Berlin: Springer-Verlag, pp. 47-64, 2005.

[2] M. Utiyama, and H. Isahara, "Reliable measures for aligning Japanese-English news articles and sentences," *Proc. 41st Annual Meeting of the ACL*, pp. 72-79, 2003.

[3] J. Hutchins, "Machine translation and computer-based translation tools: What's available and how it's used," *A New Spectrum of Translation Studies*, J. M. Bravo, ed. Valladolid: University of Valladolid, pp. 13-48, 2003.

[4] H. Somers, ed., *Computers and Translation: A Translator's Guide*, Amsterdam: John Benjamins, 2003.

[5] P. Brown, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79-85, 1990.

[6] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," *Proc. 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 48-54, 2003.

[7] K. Yamada, *A Syntax-Based Statistical Translation Model*, PhD thesis, University of Southern California, 2002.

[8] S. Sato, *Example-Based Machine Translation*, PhD thesis, Kyoto University, 1991.

[9] J. Hutchins, "Towards a definition of example-based machine translation," *Proc. Second Workshop on Example-based Machine Translation at MT Summit X*, pp. 63-70, 2005.

[10] S. Richardson et al., "Overcoming the customization bottleneck using example-based MT," *Proc. ACL Workshop on Data-driven Machine Translation*, pp. 9-16, 2001.

[11] Y. Matsumoto et al., *Japanese Morphological Analysis System ChaSen Version 2.0 Manual*, NAIST Technical Report, NAIST-IS-TR99009, 1999.

[12] H. Liu, *MontyLingua: An End-to-End Natural Language Processor with Common Sense*, MIT Media Lab, 2004.