Incremental Learning of Transfer Rules for Customized Machine Translation

Werner Winiwarter

Faculty of Business, Economics, and Computer Science,
University of Vienna, Liebiggasse 4, A-1010 Vienna, Austria,
werner.winiwarter@univie.ac.at,
WWW home page: http://www.ifs.univie.ac.at/~ww/

Abstract. We present a machine translation system, which translates Japanese into German. We have developed a transfer-based architecture in which the transfer rules are learnt incrementally from translation examples provided by the user. This means that there are no handcrafted rules, but, on the contrary, the user can customize the system according to his own preferences. The translation system has been implemented by using Amzi! Prolog. This programming environment had the big advantage of offering sufficient scalability even for large lexicons and rule bases, powerful unification operations for the application of transfer rules, and full Unicode support for Japanese characters. Finally, the application programming interface to Visual Basic made it possible to design an embedded translation environment so that the user can use Microsoft Word to work with the Japanese text and invoke the translation features directly from within the text editor.

1 Introduction

For language students and other people interested in Japanese documents, the Web makes available a wealth of information. In general, after reaching a certain level of competence, the reading of written material represents an excellent way to improve the fluency by learning new terminology or grammatical structures with little effort. However, this approach to language acquisition, which works so well with many languages, is seriously hampered by the complexity of the Japanese writing system. Japanese texts are a mixture of the two syllable writings *hiragana* and *katakana* as well as the Japanese versions of Chinese characters called *kanji*. The two syllable writings are relatively easy to learn with only 48 different characters each, but there are several thousand, mostly quite complex kanji of which the pronunciations or *readings* often depend on the textual context. Another severe problem in Japanese is that the individual words are not separated by spaces so that the reader has to guess the word boundaries. All these difficulties make reading and translating Japanese sentences a cumbersome and tedious process. If the reader gets to an uncomprehensible text portion, he must first guess where a word starts and then consult a lexicon. The dictionary lookup is quite straightforward as long as the reader is sure about

the correct pronunciation, otherwise he has to consult a kanji dictionary, which lists kanji and their readings categorized by 214 basic elements or *radicals*. The retrieval of this kanji information is again a time-consuming task.

Online documents have the great advantage that they enable the use of convenient tools, which assist the reader in comprehending the meaning of the Japanese text. Today there are several Web sites that offer information about kanji as well as English or German translations of Japanese words as pop-up hints just by pointing with the mouse at a certain text position, e.g. `www.popjisyo.com`. Even if these tools are very useful, it must be remarked that there are often problems with the correct segmentation and the retrieval of conjugated words.

In a previous project we developed a reading tool for the use within Microsoft Word. We implemented this environment by using Amzi! Prolog, which provides full Unicode support so that Japanese characters can be used freely in the Prolog source code. Its application programming interface to Visual Basic enabled us to embed the Prolog program into the text editor. The implemented functionality of our reading tool included correct segmentation, lookup of conjugated words, and the addition of new word definitions. This application represented also an evaluation of the scalability of Amzi! Prolog. We could achieve excellent performance although we searched 6,355 entries extracted from the kanji dictionary KANJIDIC, 100,014 entries from the Japanese-English dictionary EDICT (for both see `etext.lib.virginia.edu/wwwjdic/wwwjdic.html`), and 190,251 entries from the Japanese-German dictionary WaDokuJT (`www.wadoku.de`).

Another, less satisfying observation with using our reading environment was that even with all the available information it was often still not possible to correctly reproduce the intended meaning of the Japanese text. The main reason for this lies in the complexity of the translation task for the language pair Japanese–German caused by the very different grammars of the two languages. In addition, Japanese introduces ambiguity regarding several grammatical features, e.g. there exist no articles, no inflections to indicate number or case, and only two tenses. The ambiguity is further increased dramatically by the extensive use of ellipsis in Japanese. Therefore, a machine translation system requires sophisticated disambiguation and anaphoric resolution strategies.

Instead of boring the reader with a lengthy discussion of the state of the art of systems available for Japanese translation, we show the results of an amusing experiment in Fig. 1. The figure lists the attempts of several machine translation programs to translate a sentence about producing a parchment codex. We could only find one translation program into German, all others are into English. All the examples are taken from free online translation Web sites, except the last entry, which was produced by a commercial product. We encourage the reader to participate in the entertaining mental exercise to figure out the intended meaning of the original sentence (the solution can be found in Fig. 3).

This unsatisfactory situation was enough motivation for us to meet the challenge of developing a high quality machine translation system from Japanese into German. In our approach the system learns the transfer rules incrementally

Japanese sentence:
これは、片面だけに字を書いて、同じ大きさに切りそろえたものを、何枚も革のひもでとじた。

Machine translation by WorldLingo (www.worldlingo.com/products_services/worldlingo_translator.html):
Dieses, den Brief auf gerade die eine Seite, die schreibend, die sie in die gleiche Größe trimmt, geschlossen vielen mit der Zeichenkette des Leders.
(≈ This, the letter at just the one side, which writing, which it trims in the same size, closed many with the character string of the leather.)

Machine translation by SYSTRAN (www.systransoft.com/):
This, writing the letter on just the one side, those which it trims in the same size, closed many with the string of the leather.

Machine translation by Excite (www.excite.co.jp/world/url/):
This is leather many sheets about what wrote the character only to one side and was cut to an even length in the same size. With a string It closed.

Machine translation by @nifty (www.nifty.com/globalgate/):
This wrote the character only to one side and also closed many things cut to an even length in the same size with the string of leather.

Machine translation by TransLand (www.brother.co.jp/jp/honyaku/demo/index.html):
A letter was written only to the settlement side, and this じ how many sheets of things which cut it into the same size and which was completed with the leather string, too.

Machine translation by iTranslator (itranslator.mendez.com/BGSX/BGSXeng_us-EntryPage.htm):
For this, as writing a character, I cut and leveled it for a similar size for one side only, and any sheets are ひもでとじた of a leather.

Machine translation by 訳せ!!ゴマ (ai2you.com/goma/):
This bound the one, that writes only to one side and evenly cut a/the character to the same size with the strings of many sheets of leather.

Fig. 1. Example output of machine translation systems

from translation examples so that the user can customize the system according to his personal preferences.

The rest of the paper is organized as follows. In Sect. 2 we first provide a brief discussion of related work. Then we give an overview of the system architecture in Sect. 3 before we describe the technical details of the individual components of our translation environment in Sect. 4, i.e. tokenization, parsing, learning, transfer, and generation. Finally, we close the paper with concluding remarks and an outlook on future work.

2 Related Work

Research on machine translation has a long tradition (for good overviews see [8, 4–6, 9]). 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. For more general use only systems for very similar language pairs promise to produce output that is acceptable or at least comprehensible. 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 [7]. This is true for transfer-based translation systems,

which try to find mappings between specific language pairs, and even more so for interlingua-based systems aiming to find a language-independent representation that mediates among arbitrary languages. The most ambitious initiative in this direction is probably UNL (`www.undl.org`); one recent system limited to the translation of Japanese, Spanish, and Arabic texts into English is GAZELLE [3].

It is very disappointing to have to notice that the translation quality has not much improved in the last 10 years [10]. One main obstacle on the way to achieving better quality is seen in the fact that most of the current machine translation systems are not able to learn from their mistakes. 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 translation results is usually lost for future translations.

As a solution to this bottleneck, example-based machine translation tries to learn the transfer rules automatically on the basis of large bilingual corpora for the language pair. Whereas early systems relied on purely statistical techniques [1], recent work focusses on hybrid approaches that make also use of linguistic knowledge [12, 11, 2]. However, the achieved results are so far only of fair quality and still a long way from high quality machine translation.

3 System Architecture

In our approach we use translation examples provided by the user to learn the transfer rules incrementally by aligning the corresponding syntax trees. There were several reasons for this design choice:

- we did not have the resources to manually build the transfer rule base, also we consider it as difficult to extend such a rule base in a consistent way,
- as pointed out in Sect. 2 the results of existing example-based approaches are not sufficient for high quality translation so that we decided not to rely on inaccurate heuristics,
- whereas there exist quite many large bilingual corpora for Japanese–English, we had no such linguistic resources available for Japanese–German,
- in our opinion there exists no "perfect" translation but only a preferred one for a certain user, therefore we aim at full customization of our system,
- the structured representation in the syntax trees proved to be an efficient input to the learning algorithm, and we can display the trees as additional valuable information for language students.

Figure 2 shows the architecture of our translation system. For learning new transfer rules we use a Japanese–German sentence pair as input. Both sentences are first analyzed by the *tokenizer* modules, which produce the correct segmentations into word tokens associated with their part-of-speech (POS) tags. Both token lists are then transformed into syntax trees by the *parsing* modules. The syntax trees represent the input to the *learning* module, which uses a tree matching algorithm to derive new transfer rules.

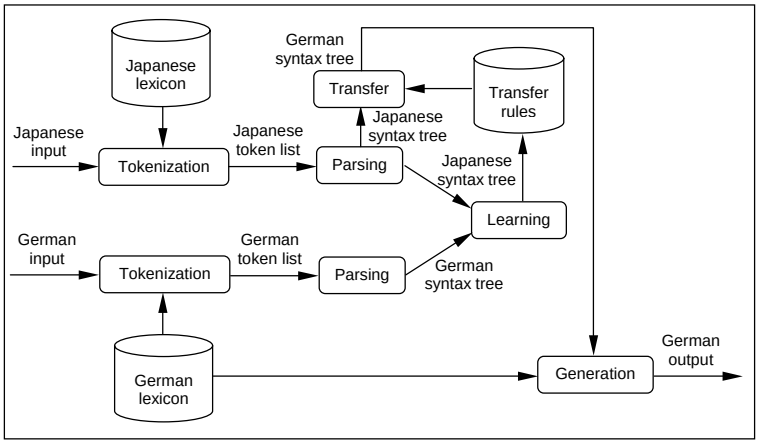


Fig. 2. System architecture

To translate a Japanese sentence we invoke the *transfer* module. It applies the transfer rules stored in the rule base to transform the Japanese syntax tree into the corresponding German syntax tree. Finally, the task of the *generation* module is to produce the surface form of the German sentence as a character string. Of course, a translated sentence can again be corrected by the user to incrementally improve the quality of the transfer rule base.

In the following section we provide some more technical details about the individual modules. We illustrate the mode of operation by using the sentence in Fig. 1 as a running example throughout the rest of this paper.

4 System Description

4.1 Tokenization

Because there are no delimiters in Japanese to segment individual words, we have to treat a sentence as a single string by using the `findall` predicate to retrieve all words from the Japanese lexicon that are left substrings. Japanese has a rich system of conjugation for verbs and adjectives, therefore we have to consider all concatenations of word stems and endings for these classes. From the set of word candidates we choose the correct word by applying some disambiguation rules. The default choice is the longest matching sequence. After the successful identification of one word, we remove it from the sentence string and start the search again. This task requires quite some processing power, but is solved by Amzi! Prolog even for large lexicons without any problems.

Figure 3 shows the output of the tokenization module for our example sentence (and as promised our solution for the translation into German). It also displays the token list for the German sentence on the right side. Compared to Japanese segmentation, this task is trivial for German and can be solved by

Japanese sentence:
これは、片面だけに字を書いて、同じ大きさに切りそろえたものを、何枚も革のひもでとじた。

Roman transcription:
Kore wa, katamen dake ni ji o kaite, onaji ookisa ni kirisoraeta mono o, nanmai mo kawa no himo de tojita.

Human translation into German:
Man hat es nur einseitig beschrieben und mehrere auf gleiche Größe zurechtgeschnittene Blätter mit Lederriemen zusammengebunden.
(≈ It was written on only one side, and several sheets, trimmed to the same size, were bound together with leather laces.)

これ/dpr	demonstrative pronoun – this
は/par	particle – (topic indicator)
、/cma	comma
片面/nou	noun – one side
だけ/suf	suffix – only
に/par	particle – on
字/nou	noun – character
を/par	particle – (direct object indicator)
書く/vte	verb te-form – to write
、/cma	comma
同じ/ano	adjectival noun – same
大きさ/nou	noun – size
に/par	particle – to
切りそろえる/vta	verb ta-form – to trim
もの/nou	noun – thing
を/par	particle – (direct object indicator)
、/cma	comma
何/ipr	interrogative pronoun – what
枚/cou	counter – sheet
も/par	particle – also
革/nou	noun - leather
の/par	particle – (attribution indicator)
ひも/nou	noun – lace
で/par	particle – with
とじる/vta	verb ta-form – to bind together
。/per	period

man/npr	indefinite pronoun – one
haben/apr	auxiliary verb present tense – to have
es/pep	personal pronoun – it
nur/adv	adverb – only
einseitig/apo	adjective positive comparison – on one side
beschreiben/vpp	verb past participle – to write
und/con	conjunction – and
mehrere/npr	indefinite pronoun – several
auf/prp	preposition – to
gleich/apo	adjective positive comparison – same
Größe/nsg	noun singular – size
zurechtschneiden/vap	verb attributive past participle – to trim
Blatt/npl	noun plural – sheet
mit/prp	preposition – with
Lederriemen/nsp	noun singular or plural – leather lace
zusammenbinden/vpp	verb past participle – to bind together
. /per	period

Fig. 3. Example of token lists

simply using the predicate `string_tokens`. The demonstrative pronoun "kore" is an anaphoric reference to "the parchment", which was introduced before in the Japanese text. Some ambiguities are resolved later during parsing, e.g. for the noun "Lederriemen" plural and singular forms are identical. Of course, the token lists can be consulted by users to offer valuable information to language students.

4.2 Parsing

One interesting property of Japanese grammar is that it uses postpositions instead of prepositions and that the predicate is at the end of the sentence (see Fig. 4). Therefore, it is easier to parse a Japanese sentence from right to left. We use the Definite Clause Grammar (DCG) preprocessor of Amzi! Prolog so that we can write the grammar rules without including the linked difference lists. Instead of using a fixed structure to represent the syntax tree, we opted for a more flexible and robust representation by using *sets* modeled as Prolog lists. A sentence is a set of constituents, and each constituent is a compound term of arity 1 with the constituent name as principal functor and the argument being either a *simple constituent* (`feature_value` or `word_category/word`) or a *complex constituent* (set of subconstituents).

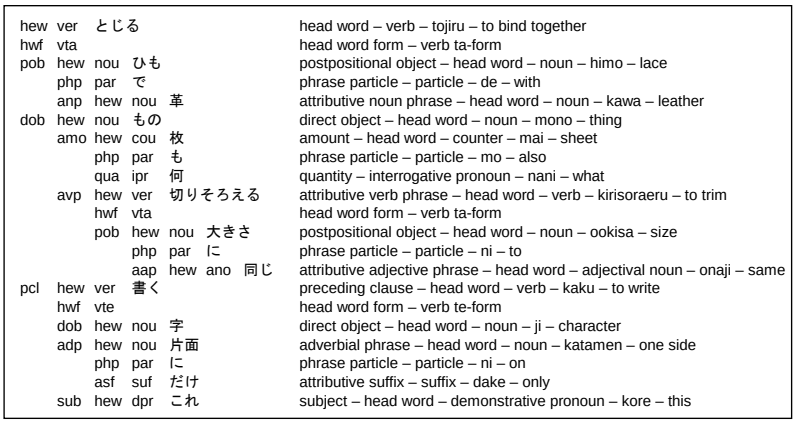


Fig. 4. Example of Japanese syntax tree

We have implemented several generic predicates to manipulate complex constituents, e.g. to find, insert, remove, or replace subconstituents. This flexible representation has the advantage that it is compact, because empty optional constituents are not stored explicitly. It allows for several subconstituents with the same constituent name, e.g. several adverbial phrases, and is not affected by the ordering of the subconstituents. The latter is important for a robust and effective realization of the transfer module so that the transfer rules can change the syntax tree without having to consider any ordering information.

For the parsing of German we proceed from left to right (see Fig. 5). As mentioned in Sect. 4.1 we resolve ambiguities for syntactic features during parsing, e.g. now we can assign the correct number plural to "Lederriemen". For the display of the parsing trees we have implemented one generic display module for both Japanese and German syntax trees, which is also able to deal with mixed representations caused by missing coverage of the transfer rule base. This proved to be a very valuable diagnosis tool, especially during the early stages of system development.

4.3 Learning

The learning module traverses the Japanese and German syntax trees and derives new transfer rules, which are added to the rule base. For that purpose we use powerful generic predicates for the simultaneous navigation in two complex constituents. We start at looking for mappings at the top level of the sentence before searching for corresponding constituents and continuing the search for finer-grained transfer rules recursively. The mapping algorithm always performs a complete traversal, new rules are learnt even if they are not required for the translation of the current sentence, in order to extract as much information as possible from the example.

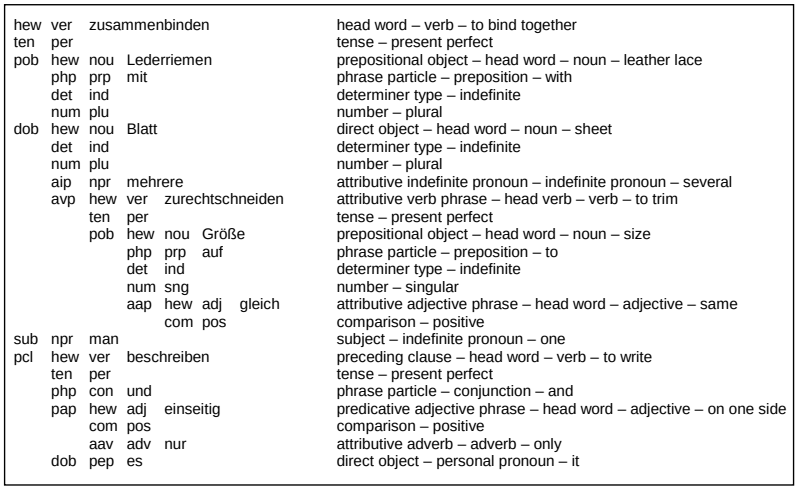


Fig. 5. Example of German syntax tree

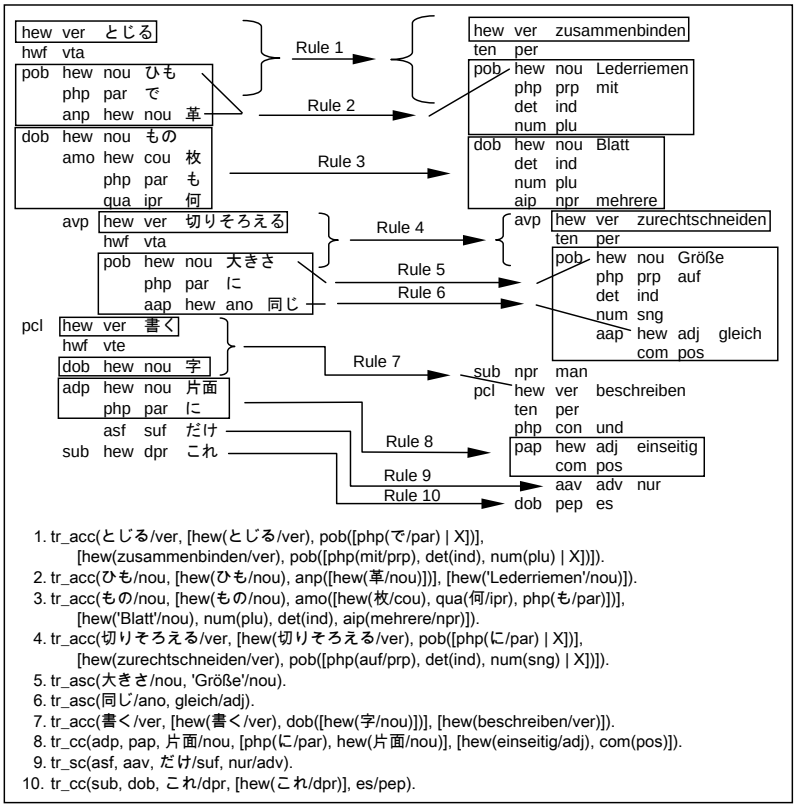
As can be seen in Fig. 6 we distinguish between 4 different types of transfer rules for simple constituents (SC) and complex constituents (CC):

- `tr_sc(C1,C2,A1,A2)`: changes the SC `C1(A1)` to `C2(A2)`,
- `tr_asc(A1,A2)`: changes the argument of an SC from `A1` to `A2`,
- `tr_cc(C1,C2,Hew,Req1,Req2)`: changes the CC `C1(A1)`, `A1`=`Req1`∪`Opt`, to `C2(A2)`, `A2`=`Req2`∪`Opt`, if `hew(Hew)`∈`A1`,
- `tr_acc(Hew,Req1,Req2)`: changes the argument of a CC from `A1`=`Req1`∪`Add`, to `A2`=`Req2`∪`Add` if `hew(Hew)`∈`A1`.

`Hew` serves as index for the fast retrieval and the reduction of the number of rules that have to be analyzed. For `tr_acc` any additional constituents are allowed as `Add`, whereas the use of `Opt` in `tr_cc` is restricted to certain optional constituents, e.g. the suffix "dake" in Rule 8 in Fig. 6. In rules of type `tr_cc` and `tr_acc` `Req1` and `Req2` can contain shared variables for unification (see Fig. 6 Rule 1 and Rule 4). In addition to these 4 general rule types we also learn a few more specific types of rules, e.g. for the correct translation of conjunctions.

4.4 Transfer

The transfer module traverses the Japanese syntax tree and searches for transfer rules that can be applied. As mentioned in Sect. 4.3, especially the condition part for rule type `tr_acc` involves sophisticated unification over two sets of constituents, i.e. the ordering of the constituents must not influence the result. Therefore, we first lookup each element of `Req` in `A1` to create a list of constituents in the same order, which can then be easily unified. As a byproduct of this sorting process we get the set difference `Add`=`A1`\`Req` as all remaining constituents in `A1` that were not retrieved.



1. tr_acc(とじる/ver, [hew(とじる/ver), pob([php(で/par) | X])],
   [hew(zusammenbinden/ver), pob([php(mit/prp), det(ind), num(plu) | X])]).
2. tr_acc(ひも/nou, [hew(ひも/nou), anp([hew(革/nou)])], [hew('Lederriemen'/nou)]).
3. tr_acc(もの/nou, [hew(もの/nou), amo([hew(枚/cou), qua(何/ipr), php(も/par)])],
   [hew('Blatt'/nou), num(plu), det(ind), aip(mehrere/npr)]).
4. tr_acc(切りそろえる/ver, [hew(切りそろえる/ver), pob([php(に/par) | X])],
   [hew(zurechtschneiden/ver), pob([php(auf/prp), det(ind), num(sng) | X])]).
5. tr_asc(大きさ/nou, 'Größe'/nou).
6. tr_asc(同じ/ano, gleich/adj).
7. tr_acc(書く/ver, [hew(書く/ver), dob([hew(字/nou)])], [hew(beschreiben/ver)]).
8. tr_cc(adp, pap, 片面/nou, [php(に/par), hew(片面/nou)], [hew(einseitig/adj), com(pos)]).
9. tr_sc(asf, aav, だけ/suf, nur/adv).
10. tr_cc(sub, dob, これ/dpr, [hew(これ/dpr)], es/pep).

Fig. 6. Example of transfer rules

This flexible definition of rules enables a robust processing of the syntax tree. One rule only changes certain parts of a constituent into the German equivalent, other parts are left unchanged to be transformed later on. Thus, our transfer algorithm deals efficiently with a mixture of Japanese–German, which gradually turns into a correct German syntax tree. Finally, we apply some structural rules, e.g. to determine the correct tense, voice, mood, etc.

4.5 Generation

To generate the surface form of the German sentence, we traverse the syntax tree in the correct order and produce a list of word tokens along the way. This list is then transformed into a single character string by inserting spaces where appropriate. The main difficulty is the generation of the inflected word forms because German has a rich system of conjugation and declension. The required syntactic information is partly encoded in the syntax tree (e.g. number or tense) and partly derived from the German lexicon (e.g. gender of nouns). One particular problem is to derive the correct surface form of anaphoric references. For that

purpose we store candidates for antecedents in previous sentences; e.g. in Fig. 6 it is important to know that the personal pronoun "es" refers to the noun phrase "das Pergament" (the parchment), i.e. to an antecedent with number singular and gender neuter.

5 Conclusion

We have completed the implementation of the system and are now in the process of filling the transfer rule base with the help of several language students from the University of Vienna. So far, the feedback from the students has been very positive, sometimes even enthusiastic. For some, our system has already become an invaluable companion throughout their language studies. The bidirectional knowledge transfer fosters a lively interaction and makes the learning process more interesting and entertaining.

Whereas at the moment language students are our main target audience, we hope to reach a level of linguistic competence in the near future that will make it also possible for non-specialist users to benefit from our translation environment. In addition to constantly extending the coverage of our rule base, future work will also concentrate on a thorough evaluation of the system according to the FEMTI framework (`www.isi.edu/natural-language/mteval/`).

References

1. Brown, P.: A statistical approach to machine translation. Computational Linguistics **16(2)** (1990) 79–85
2. Brockett, C. et al.: English-Japanese example-based machine translation using abstract linguistic representations. Proc. of the COLING-2002 Workshop on Machine Translation in Asia, Taipei, Taiwan (2002)
3. Germann, U.: Making semantic interpretation parser-independent. Proc. of the 3rd AMTA Conf., Longhorne, USA (1998) 286–299
4. Hutchins, J.: Machine translation: past, present, future. Ellis Horwood (1986)
5. Hutchins, J.: Machine translation over 50 years. Histoire epistémologie langage **23(1)** (2001) 7–31
6. Hutchins, J.: Has machine translation improved? some historical comparisons. Proc. of the 9th MT Summit, New Orleans, USA (2003) 181–188
7. Hutchins, J.: Machine translation and computer-based translation tools: what's available and how it's used. In: Bravo, J. M., ed.: A new spectrum of translation studies, University of Valladolid (2003)
8. Hutchins, J., Somers, H.: An introduction to machine translation. Academic Press (1992)
9. Newton, J., ed.: Computers in translation: a practical appraisal. Routledge (1992)
10. Somers, H., ed.: Computers and translation: a translator's guide. John Benjamins (2003)
11. Watanabe, T., Imamura, K., Sumita, E.: Statistical machine translation based on hierarchical phrase alignment. Proc. of the 9th Intl. Conf. on Theoretical and Methodological Issues in Machine Translation, Keihanna, Japan (2002) 188–198
12. Yamada, K.: A syntax-based statistical translation model. PhD thesis, University of Southern California (2002)