[TMI 99: *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, August 1999, Chester, UK]

Adding Linguistic Knowledge to a Lexical Example-Based Translation System

Ralf D. Brown
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
ralf+@cs.cmu.edu

Abstract

Example-Based Machine Translation (EBMT) using partial exact matching against a database of translation examples has proven quite successful, but requires a large amount of pre-translated text in order to achieve broad coverage of unrestricted text. By adding linguistically tagged entries to the example base and permitting recursive matches that replace the matched text with the associated tag, substantial reductions in the required amount of pre-translated text can be achieved. A modest investment of time - on the order of two person-weeks - adding linguistic knowledge reduces the required example text by a factor of six or more, while retaining comparable translation quality. This reduction makes EBMT more attractive for so-called "low-density" languages for which little data is available.

1 Introduction

The example-based machine translation engine used in the Pangloss and DIPLOMAT projects (Brown 1996) has, until now, been purely lexical. Unlike other EBMT systems which include parsers and perform similarity matching on parse trees (Nagao 1984; Sato 1992), find the most similar complete sentences and modify their stored translations to generate a translation (Veale & Way 1997), or produce an optimal partition of the input (Maruyama & Watanabe 1992; Nirenburg et al. 1993), the DIPLOMAT EBMT engine performs overlapping partial exact matches, finding all occurrences within the example base of any contiguous phrase from the input to be translated. The resulting partial translations of the input are made available as candidates for use in a multi-engine machine translation system (Frederking et al. 1994). While this approach has the advantage of being very quick at run-time and very easy to provide with examples (no linguists are required to build a parser, only translators to generate examples if they are not already available), it has the drawback of requiring substantial amounts of pre-translated text - several million words for good coverage of unrestricted texts.

The question arises whether it is possible to reduce the amount of example text required by providing a modest amount of linguistic information in order to permit generalization of the examples. For instance, if one can identify noun phrases in the source-language text, one can then (within limits) substitute any other noun phrase wherever a noun phrase occurs. Ideally, one would not need expert linguists to create a full grammar of the language – a small subset grammar capturing the most frequent phenomena, generated by a non-linguist, should suffice. As this paper shows, that indeed proves to be the case.

2 Method

The basic matching algorithm is to search the example base for contiguous occurrences of successive words in the input to be translated, using an inverted index which lists all occurrences for each unique word. In this manner, the largest phrases from the input which are contained in each sentence in the example base are found. For each match, the corresponding translation is determined by performing a word-level alignment (Brown 1996; Brown 1997) of the two halves of the translation example (a process which may fail or produce alignments which are deemed too poor to be usable).

This partial exact matching is extended by allowing equivalence classes. Certain words can be used interchangeably, forming an equivalence class such as numbers, weekdays, or country names. Substituting any other member of the equivalence class yields a well-formed (though occasionally semantically anomalous) sentence.

Equivalence classes are applied by replacing any matching words or phrases with the name of the equivalence class, appending a disambiguating number if that equivalence class has already been used (referred to as tokenizing in the remainder of this paper). The process is repeated until no more replacements are possible, at which time a partial exact match against the example base is performed, just as previously without equivalence classes. Since the example base has also had all members of equivalence classes replaced by the class name (tokenized), this allows interchangeable use of members of an equivalence class.

In the input to be translated, words and phrases belonging to an equivalence class are *always* replaced by the class name. In the example base, the class members are only replaced if an appropriate translation is present in the target-language half of the example. To permit proper matching against the example base, ambiguous "words" are permitted which match any one of several alternatives at that location. Whenever a single word is replaced by its class name, the original word is retained as an alternative for matching; unfortunately, this is not possible for phrases as the difference in length would cause erroneous matches when examining the index. This capability for ambiguous terms also allows words to be in multiple equivalence classes provided that the translations are mutually distinct. (If there were a common translation between different equivalence classes, the system would be unable to decide which to use when indexing the example base.)

Whenever a term is replaced by its class name, the corresponding translation is remembered. Once a translation of the tokenized text has been found, each token is expanded by substituting the translation which was remembered when the text was initially tokenized. This back-substitution step yields the final translation which is output, and is what makes equivalence classes work. See Figure 2 for an example of back-substitution, which is described in more detail below.

As an example, consider the sentence

```
John Miller flew to Frankfurt on December 3rd.
```

This becomes

```
<firstname-m> <lastname> flew to <city> on <month> <ordinal>.
```

after an initial tokenization pass, and then

```
<person-m> flew to <city> on <date>.
```

```
;;;(TOKEN <NOUN-M>)                    ;;;(TOKEN <NP-M>)
book                                   the <NOUN-M>
livre                                  le <NOUN-M>

;;;(TOKEN <NP-M>)                      ;;;(TOKEN <NP-F>)
<POSS> <ADJ-N> <NOUN-M>                the <NOUN-F>
<POSS> <NOUN-M> <ADJ-M>                la <NOUN-F>

;;;(TOKEN <NP-M>)                      ;;;(TOKEN <NP-F>)
<POSS> <NOUN-M>                        a <COLOR> <NOUN-F>
<POSS> <NOUN-M>                        une <NOUN-F> <COLOR>
```

Figure 1: Sample Production Rules

after a second pass. The tokenized form will now match

```
Dr.  Howard Johnson flew to Ithaca on 7 April 1997.
```

among many other possibilities.

A further generalization of equivalence classes involves repeated (recursive) matching against the example base. For this extension, translation pairs in the example base are tagged with a token which preferably contains linguistic information such as gender and number. Tagged entries are not limited to literal strings – they may themselves contain tokens, allowing the use of paired production rules to create a grammar, as shown in Figure 1.

To perform a translation, the system first searches for phrases that completely match one or more tagged entries, and then substitutes the associated tags into the input text (see Figure 3 for the result of such substitution on a translation example). This process is repeated until there are no more complete matches of tagged entries, at which point an extended form of the normal partial-exact match against all examples – including tagged entries – in the knowledge base is performed. As is the case when using equivalence classes, at each step the appropriate back-substitution is remembered so that it can be applied to the tokenized translation in order to produce the final output.

The process of matching against the corpus is more complex when grammar rules are involved, because not all alternative terms which will be matched represent the same number of words in the input. Each of the individual substitutions produced at any stage of the repeated tokenization described above may be matched against the corpus. To accomplish this, G-EBMT sequentially processes each position i in the original input, looking for adjacent occurrences of any of the tokens *ending* at position $i - 1$, the original word w_{i-1}, or any not-yet-completed matches ending at $i - 1$, with w_i or any of the tokens *starting* at position i. The difficult part is properly tracking the starting and ending positions of each match in the original input in addition to the length of the actual match using tokenized text.

Recursive matching permits a word to be in multiple equivalence classes even when the translations are not distinct, and can be applied more generally than simple tokenization because replacements are only made in the proper context.

```
... the affordable painters ...
  ==> ... the (<adj-s> <adj-p>) <noun-m-p> ...
  ==> ... the <adj-p> <noun-m-p> ...
  ==> ... <np-m> ...
        -=- translation into Spanish -=-
  ==> ... <np-m> ...
  ==> ... los <noun-m-p> <adj-p> ...
  ==> ...  los pintores accesibles ...
```

Figure 2: Disambiguation through Linguistic Constraints

When the tags contain linguistic information, this information can be used to enforce constraints and thus select the appropriate sense (and translation) of a word. For the example shown in Figure 2, the English word “affordable” can be translated as either a singular or plural adjective; this is indicated by showing all alternatives for a given word as a list in parentheses. After a first recursive matching pass, both “affordable” and “painters” are tokenized. Searching the corpus for a further tagged match of this initial result yields only the masculine noun phrase in which both adjective and noun are plural, which disambiguates “affordable” as the plural form. Once no more matches are possible, the translation of the fully-tokenized input is determined, and the tokenization is reversed by back-substituting the appropriate translation for each tokenized term. The final result is a translation in which the correct alternative has been selected.

Back-substitution is implemented by maintaining a list of replacements which were made in the process of tokenizing the input, as well as during recursive matching. The replacement list indicates for each token the position of that token, the source-language word or phrase it replaces, and the translation that was associated with the matched source-language text. To convert a token in the output into the proper translation, simply find the identical token (including the disambiguating number, if present) in the replacement list, and substitute the indicated target-language string. While Figure 2 shows the initial tokenization being reversed step-by-step, the actual implementation uses the equivalent method of immediately back-substituting any tokens which may be present in the phrase that corresponds to a token, so that the replacement list always shows the full surface form for a token. This allows the final back-substitution to be performed in a single pass through the target-language output.

3 Experiment

The efficacy of recursive matching with linguistic information was tested for both Spanish-English and French-English translation. In each case, translations were performed three times: using only the example texts, using the example texts and equivalence classes, and using full recursive matching on both the example texts and equivalence classes.

To determine the effect of varying the corpus size, both the French and Spanish example bases were split into relatively small sections. After each section was added to

Original text:

E: Furthermore , for the reasons mentioned previously , it is hoped that the person contacted will respect the confidentiality of the inquiry .

F: En outre , on espere , pour les raisons susmentionnées , que la personne contactée respectera le caractère confidentiel de 1'enquête .

After recursive-match indexing:

E: Furthermore , for <np-m-p> mentioned previously , it is hoped <cond> <np-f> <adj-f> <v-ft> <det-m> confidentiality <pp> .

F: En outre , on espère , pour <np-m-p> susmentionnées , <cond> <np-f> <adj-f> <v-ft> <det-m> caractère confidentiel <pp> .

Figure 3: Results of Indexing Translation Example

the indexed corpus, the test text was translated and evaluated for both coverage of the input and average match length. Match length serves - in lieu of manual evaluation - as an approximate indicator of translation quality, since longer segments include more contextual information and are thus less likely to be translated using the wrong sense. Moreover, longer matches are more likely to be at least partially correct even when the word-level alignment is incorrect.

For the Spanish system, the primary source of translation examples is the UN Multilingual Corpus (Graff & Finch 1994). A total of 5397 term/translation pairs in 94 equivalence classes (the bulk of them numbers and names of various kinds, such as days of the week, months, animals, flowers, trees, minerals, chemical elements, companies, organizations, countries, honorifics, political/military ranks, given names, and surnames) were generated and manually translated. Some 450 grammar rules were produced, many derived from an early version of the French grammar rules. To produce a set of entries with morphological information, a file of Spanish words and their morphological analysis that had been produced during the Pangloss project was translated word-by-word using the EBMT system's dictionary and then semi-automatically filtered. The filtering consisted of searching for words which had a combination of part-of-speech/tense/number for the Spanish word with an appropriate suffix for the English translation (e.g. *-able* for adjectives, *-ed* for past-tense verbs, etc.) and performing a quick manual cleaning pass.

Two collections of testing text were used for Spanish: 276 held-out sentences (9089 words) from the UN corpus to gauge “in-domain” performance, and 253 sentences (9167 words) of newswire text from a 1993 ARPA machine translation evaluation (Frederking et al. 1993) to check “out-of-domain” performance. Only the in-domain performance will be presented in this paper; the out-of-domain results peak 8 to 10 percentage points lower (with the smaller difference occurring when using generalization).

For the French system, the primary source of translation examples is the IBM Hansard corpus (Linguistic Data Consortium 1997). The equivalence-class information is less extensive than that used for Spanish (4188 term/translation pairs in 98 equivalence classes), but shares 2989 proper names and non-translating templates – such as lists of numbers or countries – used by the Spanish system. While the approximately 540 grammar rules were also hand-crafted, the morphological information for more than 75,000 French words were derived from the ARTFL French-English dictionary (ARTFL Project 1998) with some semi-automatic additions (such as gender and various verb tenses) where suffixes or auxiliary verbs allowed an easy determination.

The test text for the French system consisted of 737 paragraphs (45,724 words) from the Bellcore Hansards (Linguistic Data Consortium 1997). Because the IBM and Bellcore corpora contain texts from different time periods, the test text does not contain any of the training text.

The grammar rules for both systems were written by the author, who is not a trained linguist. Starting with a variety of common English patterns such as DET-ADJ-N for noun phrases, the corpus was examined to determine the predominant translation patterns. After creating a preliminary set of grammar rules, the corpus was indexed and the indexed version examined for additional common patterns, as well as higher-order patterns that could be built from the results of previous rules. This process was iterated several times.

The effort of adding the grammar rules and linguistic information was quite modest, totalling an estimated 70-80 hours for the French system and 50-60 hours for the Spanish system. While the availability of morphological information for both French and Spanish considerably reduced the level of effort, for many language pairs much of the work can be performed automatically even without such data, given a bilingual dictionary which is required anyway. By matching suffixes or other lexical features, much as was done for the Spanish system (and to a lesser extent for the French system), many of the most frequent morphological variations can be captured.

4 Results

As was expected, adding equivalence classes produces a small but noticeable improvement, and the greater infusion of generalization due to recursive matches produces a greater improvement. Figures 4 and 5 show how coverage of the French and in-domain Spanish input increases as more text is added to the example base; in both figures, the bottom-most curve is the performance with example text only, the curve which is slightly above that one is the result of adding equivalence classes, and the top-most curve is the performance with all knowledge sources. The performance curve for recursive matching in Figure 4 starts substantially to the right of the other curves because the example base must be expanded with entries associating linguistic tags with various words - some 224,000 words of total text (including the grammar rules) for French and 43,000 words for Spanish. The slight dip near 500,000 words for Spanish is the result of a somewhat heterogeneous corpus; the first half-million words consist of non-parliamentary text (primarily newswire), so when the experiment reaches the example text which is actually in the same domain as the test text, performance receives an

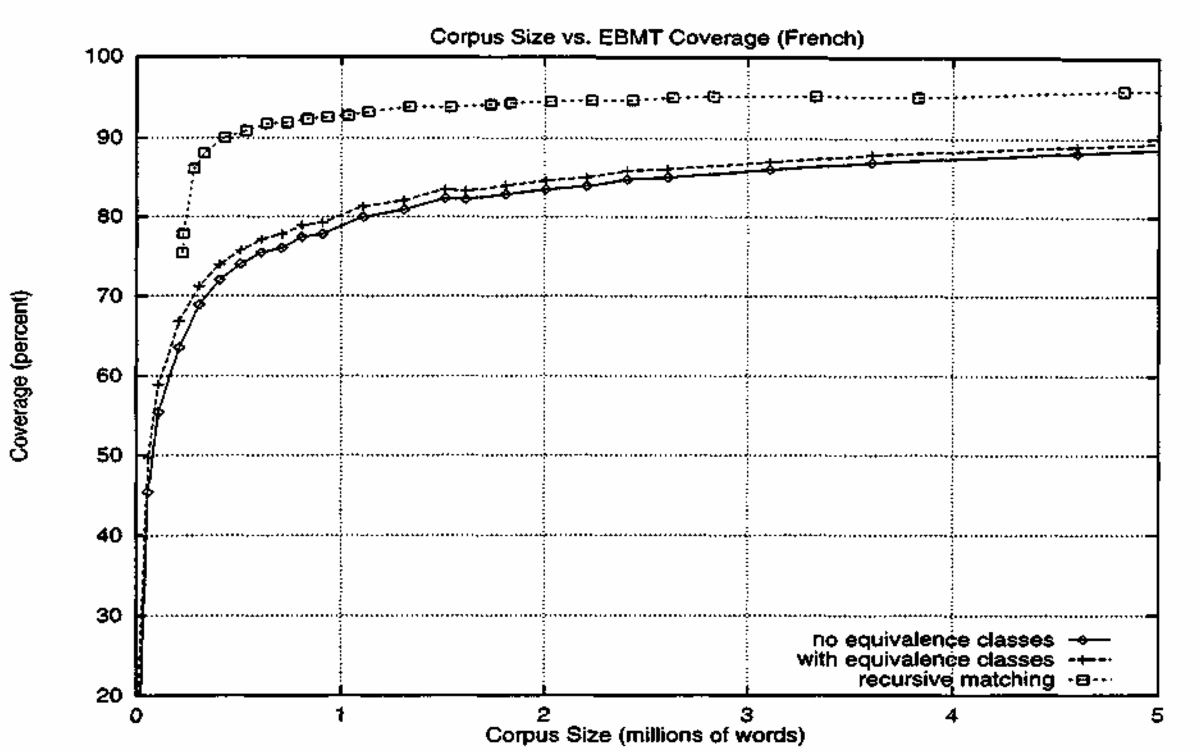

Figure 4: French-English Performance: Coverage of Input

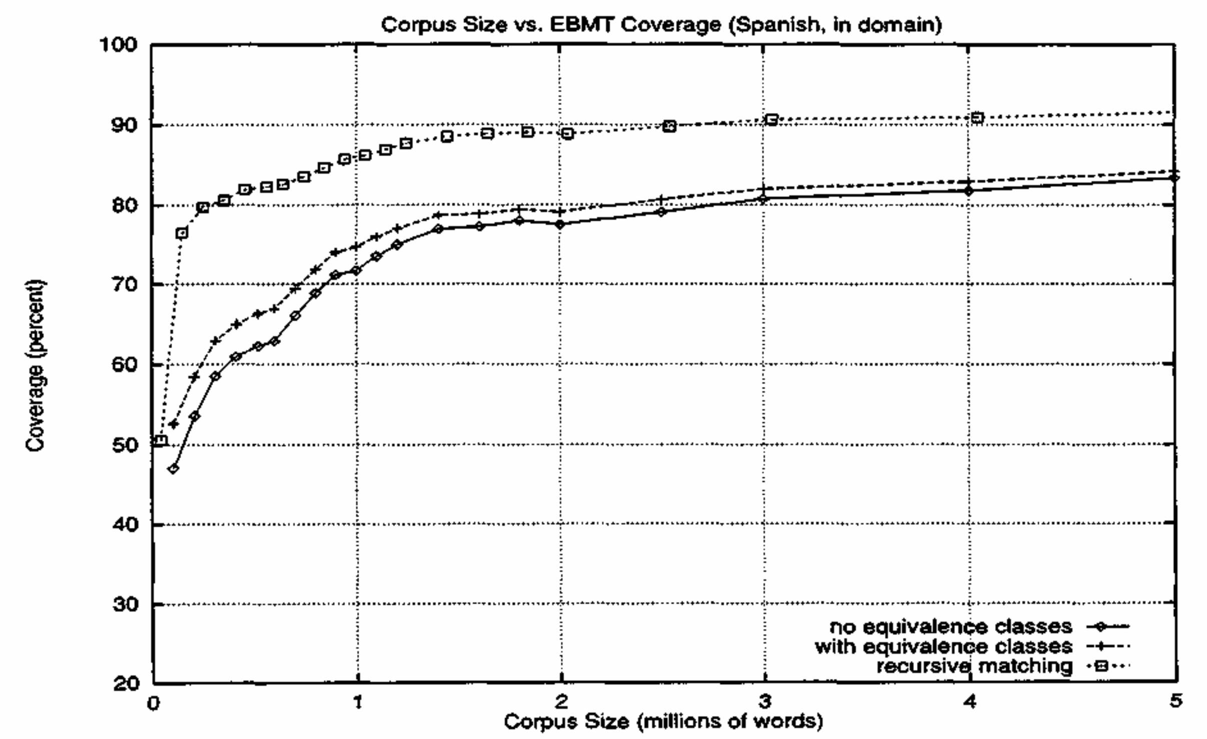

Figure 5: Spanish-English Performance: Coverage of Input

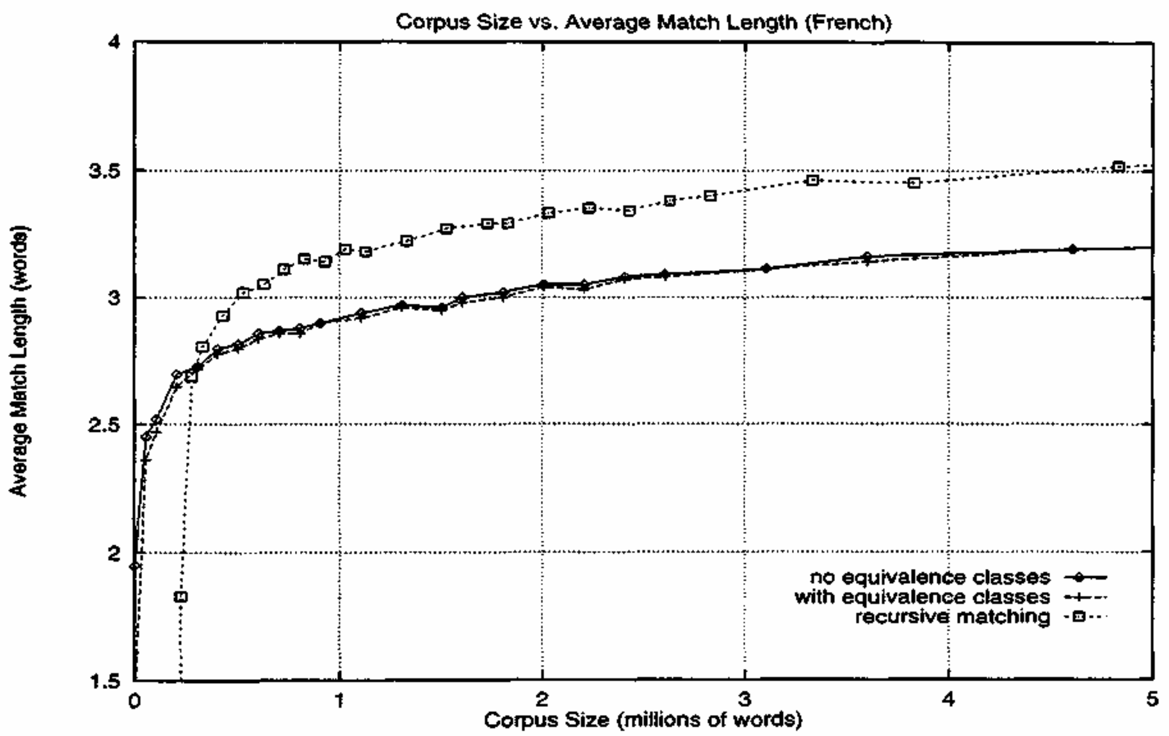

Figure 6: French-English Performance: Average Match Length

added boost.

Figure 6 shows how recursive matching also substantially increases the average length of matches against the corpus. This is to be expected, since the grammar rules allow an entire phrase to be collapsed into a single token, thus shortening the actual text being matched against the corpus. For very small corpora, the recursive matching has a shorter average match length because most matches are single-word matches against tagged morphological entries.

Overall, the amount of example text required for a given coverage of arbitrary input is reduced quite dramatically. The French system reaches 80% coverage of the test text with less than 300,000 words of training material (nearly all of which consists of grammar rules and morphological entries) when using recursive matching, but requires one million words without grammar rules and 1.2 million words when relying solely on the translation examples. Since the performance curve flattens out, the difference is even greater for higher coverage values – achieving 90% coverage requires less than half a million words with recursive matching versus 7 million words without. Similarly, the Spanish system reaches 80% coverage with only 350,000 words versus 2.5 million words and 90% with only 3 million words versus more than 11 million without generalization.

Translation quality is marginally lower when using the grammar rules, since it is easy to over-generalize. In any language, there are exceptions to the rules, for which applying the usual generalization yields incorrect translations. Many of these can be found by the indexer, because there is no corresponding translation for a matched production rule; by examining the indexer's report of such failed tokenizations, additional entries

Input:

La motion de M. Lewis est adoptée par 147 voix contre 77.
(Mr. Lewis's motion is adopted by 147 votes to 77.)

707,000-word corpus, no generalization:

```
“la motion de m . lewis” (1)      (“motion of Mr . Lewis”)
“adoptée par” (0)                 (“adopted by”)
“147 voix” (0)                    (“147 votes”)
“77 .” (0)                        (“77 .”)
```

281,000-word corpus, *with* generalization:

```
“la motion de m .” (24)             (“motion for Mister Towers”)
“est adoptee par 147” (1.33)        (“is adopted by 147”)
“par 147 voix contre 77 .” (1)     (“by 147 voice against 77 .”)
```

Figure 7: Comparison With and Without Generalization

or rules can be produced to cover these cases.[1] A further cause of reduced quality is that generalizations only produce a single, preferred translation, rather than a number of closely related translations that may vary according to context.

Figure 7 illustrates the effect of generalization on the system's output. For this example, G-EBMT was told to use very terse output: only the very best-scoring translation for any match is shown, and no matches which are entirely contained within another, larger match are shown. In normal operation, the output for even a very short sentence spans multiple pages. In Figure 7, the system output is shown in two columns; the left-hand column indicates which portion of the input was matched and the penalty score for the generated translation (zero is perfect, and five times the match length is the cut-off for output), while the right-hand column shows the translation.

The example illustrates the various effects previously discussed. With less than half as much parallel text (even counting the morphological and grammar entries), the generalized version covers more of the input, with generally longer matches, but also with lower quality. Thus, “voix” is translated as “voice” rather than “votes” because the generalization rules do not take the nature of the corpus (parliamentary proceedings) into account. The much smaller match without generalization, on the other hand, correctly uses “votes” because that is what actually appears in the corpus.

5 Conclusion and Future Work

As has been shown, the corpus of translation examples can effectively be increased by a factor of six or more with a rather modest effort. This effort is worthwhile even if a substantial amount of pre-translated text is already available, since coverage and match length still increase considerably with ten million or more words of parallel text. Further, the additional effort of adding linguistic information does not require an

[1] As of this writing, it is not yet possible for a specific example to completely override a generalization, but both the specific example and the tokenized generalization will be found when matching against the corpus.

expert linguist, but can most likely be undertaken by the same translators who provide examples for the training corpus. While increasing the effectiveness of the available translation examples is an interesting result for major languages such as French and Spanish, it is *vital* for languages which have little or no available parallel text; for such languages, being able to generalize the examples which can be found or manually generated may be what makes a translation system feasible at all.

Recursive matching works well, but currently requires a substantial investment of time in creating morphological entries if that information is not already available in electronic form. It should be possible to automatically generate morphological entries by finding other words which occur in equivalent contexts to already-labeled words, given the grammar rules and translation examples. This would permit a relatively small seed set of morphological entries to be used, reducing the manual effort. Similarly, a tool could be built to assist in the creation of grammar rules, reducing the time and minimizing the possibility of errors and omissions.

6 Acknowledgements

This research was supported by the National Science Foundation under the STIMULATE program (Grant No: IIS-9618941).

The author would like to thank Maxine Eskanazi for her assistance with the French language and the anonymous reviewers for their helpful comments.

References

ARTFL Project: 1998, *ARTFL Project: French-English Dictionary,* Project for American and French Research on the Treasury of the French Language, **http://humanities.-uchicago.edu/ARTFL.html.**

Brown, Ralf D.: 1996, 'Example-Based Machine Translation in the PANGLOSS System', in *Proceedings of the Sixteenth International Conference on Computation Linguistics,* pp. 169-174, Available at **http: //www.cs.emu.edu/~ralf/papers.html**.

Brown, Ralf D.: 1997, 'Automated Dictionary Extraction for "Knowledge-Free" Example-Based Translation', in *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-97),* Available at **http://-www.cs.emu.edu/~ralf/papers.html.**

Frederking, R., S. Nirenburg, D. Farwell, S. Helmreich, E. Hovy, K. Knight, S. Beale, C. Domashnev, D. Attardo, D. Grannes & Ralf Brown: 1994, 'Integrating Translations from Multiple Sources within the PANGLOSS Mark III Machine Translation', in *Proceedings of the first conference of the Association for Machine Translation in the Americas,* Columbia, MD.

Frederking, Robert et al.: 1993, 'An MAT Tool and Its Effectiveness', in *Proceedings of the DARPA Human Language Technology Workshop,* Princeton, New Jersey.

Graff, D. & R. Finch: 1994, 'Multilingual Text Resources at the Linguistic Data Consortium', in *Proceedings of the 1994 ARPA Human Language Technology Workshop,* Morgan Kaufmann.

Linguistic Data Consortium: 1997, *Hansard Corpus of Parallel English and French,* Linguistic Data Consortium, **http://www.ldc.upenn.edu/.**

Maruyama, H. & H. Watanabe: 1992, 'Tree Cover Search Algorithm for Example-Based Translation', in *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation,* pp. 173-184, (TMI-92).

Nagao, M.: 1984, 'A Framework of a Mechanical Translation between Japanese and English by Analogy Principle', in A. Elithorn & R. Banerji (eds), eds., *Artificial and Human Intelligence,* NATO Publications.

Nirenburg, Sergei, Constantine Domashnev & Dean J. Grannes: 1993, 'Two Approaches to Matching in EBMT', in *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation,* (TMI-93).

Sato, S.: 1992, 'CTM: An Example-Based Translation Aid System Using the Character-Based Best Match Retrieval Method', in *Proceedings of COLING-92.*

Veale, Tony & Andy Way: 1997, 'Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation', in *Proceedings of the NeMNLP'97, New Methods in Natural Language Processing,* Available at **http://www.compapp.dcu.ie/-tonyv/papers/gaijin.html.**