Automated Generalization of Translation Examples

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

Abstract

Previous work has shown that adding generalization of the examples in the corpus of an example-based machine translation (EBMT) system can reduce the required amount of pre-translated example text by as much as an order of magnitude for Spanish-English and French-English EBMT. Using word clustering to automatically generalize the example corpus can provide the majority of this improvement for French-English with no manual intervention; the prior work required a large bilingual dictionary tagged with parts of speech and the manual creation of grammar rules. By seeding the clustering with a small amount of manually-created information, even better performance can be achieved. This paper describes a method whereby bilingual word clustering can be performed using standard *monolingual* document clustering techniques, and its effectiveness at reducing the size of the example corpus required.

1 Introduction

Example-Based Machine Translation (EBMT) relies on a collection of textual units (usually sentences) and their translations. New text to be translated is matched against the source-language half of the collection, and the corresponding translations from the target-language half are used to generate a translation of the new text.

Experience with several language pairs has shown that producing an EBMT system which provides reasonable translation coverage of unrestricted texts using simple textual matching requires on the order of two million words of pre-translated texts (one million words in each language); if either language is highly inflecting, polysynthetic, or (worse yet) agglutinative, even more text will be required. It may be difficult, time-consuming, and expensive to obtain that much parallel text, particularly for lesser-used language pairs. Thus, if one wishes to develop a new translator rapidly and at low cost, techniques are needed which permit the EBMT system to perform just as well using substantially less example text.

Both the Gaijin EBMT system by Veale and Way (1997) and the author's EBMT system (1999) convert the examples in the corpus into templates against which the new texts can be matched. Gaijin variablizes the well-formed segment mappings between source and target sentences that it is able to find, using a closed set of markers to segment the input into phrases. The author's system performs its generalization using equivalence classes (both syntactic and semantic) and a production-rule grammar. First, any occurrences of terms contained in an equivalence class are replaced by a token giving the name of the equivalence class, and then the grammar rules are used to replace patterns of words and tokens by more general tokens (such as <NP> for noun phrases). (Brown, 1999) showed that one can reduce the corpus size by as much as an order of magnitude in this way.

Given that explicit, manually-generated equivalence classes reduce the need for example text, an obvious extension would be to attempt to generate these classes automatically from the corpus of pre-translated examples. This paper describes one approach to automated extraction of equivalence classes, using clustering techniques.

The remainder of this paper describes how to perform bilingual word clustering using standard *monolingual* document clustering techniques by converting the problem space; the various clustering algorithms which were investigated; and the effectiveness of generalization using the derived clusters at reducing the required amount of example text.

2 Converting the Problem

The task of clustering words according to their occurrence patterns can be restated as a standard document-clustering task by converting the problem space. For each unique word to be clustered, create a pseudo-document containing the words of the contexts in which that word appears, and use the word itself as the document identifier. After the pseudo-documents are clustered, retrieving the identifier for each document in a particular cluster produces the list of words occurring in sufficiently similar contexts to be considered equivalent for the purposes of generalizing an EBMT system.

By itself, this approach only produces a monolingual clustering, but we require a bilingual clustering for proper generalization since different senses of a word will appear in differing contexts. The method of Barrachina and Vilar (1999) provides the means for injecting bilingual information into the clustering process.

Using a bilingual dictionary — which may be created from the corpus using statistical methods, such as those of Peter Brown *et al* (1990) or the author's own previous work (Brown, 1997) — and the parallel text, create a rough mapping between the words in the source-language half of each translation example in the corpus and the target-language half of that example. Whenever there is exactly one possible translation candidate listed for a word by the mapping, generate a bilingual word pair consisting of the word and its translation. This word pair will be treated as an indivisible token in further processing, adding bilingual information to the clustering process. Forming pairs in this manner causes each distinct translation of a word to be treated as a separate sense; although translation pairs do not exactly correspond to word senses, pairs can be formed without any additional knowledge sources and are what the EBMT system requires for its equivalence classes.

For every unique word pair found in the previous step, we accumulate counts for each word in the surrounding context of its occurrences. The context of an occurrence is defined to be the N words immediately prior to and the N words immediately following the occurrence; N currently is set to 3. Because word order is important, counts are accumulated separately for each position within the context, i.e. for $N = 3$, a particular context word may contribute to any of six different counts, depending on its location relative to the occurrence. Further, as the distance from the occurrence increases, the surrounding words become less likely to be a true part of the word-pair's context, so the counts are weighted to give the greatest importance to the words immediately adjacent to the word pair being examined. Currently, a simple linear decay from 1.0 to $\frac{1}{N}$ is used, but other decay functions such as the reciprocal of the distance are also possible. The resulting weighted set of word counts forms the above-mentioned pseudo-document which is converted into a term vector for cosine similarity computations (a standard measure in information retrieval, defined as the dot product of two term vectors normalized to unit length).

If the clustering is seeded with a set of initial equivalence classes (which will be discussed below), then the equivalences will be used to generalize the contexts as they are added to the overall counts for the word pair. Any words in the context for which a unique correspondence can be found (and for which the word and its corresponding translation are one of the pairs in an equivalence class) will be counted as if the name of the equivalence class had been present in the text rather than the original word. For example, if days of the week are an equivalence class, then "did he come on Friday" and "did he leave on Monday" will yield identical context vectors for "come" and "leave", making it easier for those two terms to cluster together.

To illustrate the conversion process, consider the French word "cinq" in two examples where it translates into English as "five" (thus forming the word pair "cinq_five"):

<NUL> <NUL> Le cinq jours depuis la
<NUL> <NUL> The five days since the

elles commenceront en cinq jours . <NUL>
they will begin in five days . <NUL>

where <NUL> is used as a placeholder when the word pair is too near the beginning or end of the sentence for the full context to be present. Note that the word order on the target-language side is not considered when building the term vector, so it need not be the same as on the source-language side; the examples were chosen with the same word order merely for clarity.

The resulting term vector for "cinq_five" is as follows, where the numbers in parentheses indicate the context word's position relative to the word pair under consideration:

Word	Occur	Weight
<NUL>(-3)	1	0.333
elles(-3)	1	0.333
<NUL>(-2)	1	0.667
commenceront(-2)	1	0.667
Le(-1)	1	1.000
en(-1)	1	1.000
jours(1)	2	2.000
depuis(2)	1	0.667
.(2)	1	0.667
la(3)	1	0.333
<NUL>(3)	1	0.333

Term vectors such as the above are then clustered to determine equivalent usages among words.

3 Clustering Approaches

A total of six clustering algorithms have been tested; three variants of group-average clustering and three of agglomerative clustering. Incremental group-average clustering was implemented first, to provide a proof of concept, before the computationally more expensive agglomerative (bottom-up) clustering was implemented.

The incremental group-average algorithms all examine each word pair in turn, computing a similarity measure to every existing cluster. If the best similarity measure is above a predetermined threshold, the new word pair is placed in the corresponding cluster; otherwise, a new cluster is created. The three variants differ only in the similarity measure employed:

1. cosine similarity between the pseudo-document and the centroid of the existing cluster (standard group-average clustering)
2. average of the cosine similarities between the pseudo-document and all members of the existing cluster (average-link clustering)
3. square root of the average of the squared cosine similarities between the pseudo-document and all members of the existing cluster (root-mean-square modification of average-link clustering)

These three variations give increasingly more weight to the nearer members of the existing cluster.

The bottom-up agglomerative algorithms all function by creating a cluster for each pseudo-document, then repeatedly merging the two clusters with the highest similarity score until no two clusters have a similarity score exceeding a predetermined threshold. The three variants again differ only in the similarity measure employed:

1. cosine similarity between cluster centroids (standard agglomerative clustering)
2. average of cosine similarity between members of the two clusters (average-link)
3. maximal cosine similarity between any pair of members of the two clusters (single-link)

For each of the variations above, the predetermined threshold is a function of word frequency. Two words which each appear only once in the entire training text and have a high similarity score are more likely to have appeared in similar contexts by coincidence than two words which each appear in the training text fifty times.

Frequency	Threshold
1	1.00
2	0.85
3	0.80
4	0.75
5	0.70
6	0.65
7	0.60
8 - 9	0.55
10 - 11	0.50
12 - 15	0.45
$\geq$16	0.40

Figure 1: Clustering Threshold Function

For example, when using three words on either side as context, and a linear decay in term weights, two singleton words achieve a similarity score of 0.321 (1.000 is the maximum possible) if just one of the immediately adjacent words is the same for both, even if none of the other five context words are the same. As the number of occurrences increases, the contribution to the similarity score of individual words decreases, making it less likely to encounter a high score by chance. Hence, we wish to set a stricter threshold for clustering low-frequency words than higher-frequency words.

The threshold function is expressed in terms of the frequency of occurrence in the training texts. For single, unclustered word pairs, the frequency is simply the number of times the word pair was encountered. When performing group-average clustering, the frequency assigned to a cluster is the sum of the frequencies of all the members; for agglomerative clustering, the frequency is the sum when using centroids and the maximum frequency among the members when using the average or nearest-neighbor similarity. The value of the threshold for a given pair of clusters is the value of the threshold function at the *lower* word frequency. Figure 1 shows the threshold function used in the experiments whose results are reported here; clustering is only allowed if the similarity measure is above the indicated threshold value.

On its own, clustering is quite successful for generalizing EBMT examples, but the fully-automated production of clusters is not compatible with adding a production-rule grammar as described in (Brown, 1999). Therefore, the clustering process may be seeded with a set of manually-generated clusters.

When seed clusters are available, the clustering process is modified in two ways. First, the group-average approaches add an initial cluster for each seed cluster and the agglomerative approaches add an initial cluster for each word pair; these initial clusters are tagged with the name of the seed cluster. Second, whenever a tagged cluster is merged with an untagged one or another cluster with the same tag, the combination inherits the tag; further, merging two clusters with different tags is disallowed. As a result, the initial seed clusters are expanded by adding additional word pairs while preventing any of the seed clusters from themselves merging with each other.

One special case is handled separately, namely numeric strings. If both the source-language and target-language words of a word pair are numeric strings, the word pair is treated as if it had been specified in the seed class <**number**>. Word pairs *not* containing a digit in either word can optionally be prevented from being added to the <number> cluster unless explicitly seeded in that cluster. The former feature ensures that numbers will appear in a single cluster, rather than in multiple clusters. The latter avoids the inclusion of the many non-numeric word pairs (primarily adjectives) which would otherwise tend to cluster with numbers, because both they and numbers are used as modifiers.

Once clustering is completed, any clusters which have inherited the same tag (which is possible when using agglomerative clustering) are merged. Those clusters which contain more than one pseudo-document are output, together with any inherited label, and can be used as a set of equivalence classes for EBMT.

Agglomerative clustering using the maximal cosine similarity (single-link) produced the subjectively best clusters, and was used for the experiments described here.

4 Experiment

The method described in the previous two sections was tested on French-English EBMT. The training corpus was a subset of the IBM Hansard corpus of Canadian parliamentary proceedings (Linguistic Data Consortium, 1997), containing a total of slightly more than one million words, approximately half in each language. Word-level alignment between French and English was performed using a dictionary containing entries derived statistically from the full Hansard corpus, augmented by the ARTFL French-English dictionary (ARTFL Project, 1998). This dictionary was used for all EBMT and clustering runs.

The effects of varying the amount of training texts were determined by further splitting the training corpus into smaller segments and using differing numbers of segments. For each

Clust	Members
238	HISTOIRE HISTORY ÉCONOMIE ECONOMY
260	CERTAINEMENT CERTAINLY CERTAINEMENT SURELY CERTES SURELY JAMAIS NEVER PAS NOT PEUT-ÊTRE MAY PROBABLEMENT PROBABLY QUE ONLY RIEN NOTHING SÛREMENT CERTAINLY SÛREMENT SURELY VRAIMENT REALLY
348	CONSERVATEUR CONSERVATIVE CONSERVATEUR TORY DÉMOCRATIQUE DEMOCRATIC DÉMOCRATIQUE NDP LIBÉRAL LIBERAL
522	DERNIÈRES LAST DERNIÈRES PAST DERNIÈRES RECENT PROCHAINES NEXT QUELQUES FEW QUELQUES SOME
535	AVONS HAVE SOMMES ARE
1375	ÉLECTORALE CAMPAIGN ÉLECTORALE ELECTION
1386	FÉDÉRALES-PROVINCIALES FEDERAL-PROVINCIAL INDUSTRIELLES INDUSTRIAL OUVRIÈRES LABOUR
1528	FAÇON EVENT ÉVIDENCE CLEARLY ÉVIDENCE OBVIOUSLY
1563	HOMMES POLITICIANS PRISONNIERS PRISONERS
1652	RETOUR BACK REVENIR BACK
2008	CONVENU AGREED SIGNÉ SIGNED VU SEEN
2182	AGRICOLE AGRICULTURE ENTIER AROUND ENTIER THROUGHOUT OCCIDENTAL WESTERN
2472	AVEUGLES BLIND CHAUSSURES SHOES CONSTRUCTEURS BUILDERS PENSIONNÉS PENSIONERS RETRAITÉS PENSIONERS VÊTEMENTS CLOTHING
3539	POISSON FISH PORC PORK

Figure 2: Sample Clusters

run using clustering, the first K segments of the corpus are concatenated into a single file, which is used as input for both the clustering program and the EBMT system. The clustering program is run to determine a set of equivalence classes, and these classes are then provided to the EBMT system along with the training examples to be indexed. Held-out Hansard text (approximately 45,000 words) is then translated, and the percentage of the words in the test text for which the EBMT system could find matches and generate a translation is determined.

To test the effects of adding seed clusters, a set of initial clusters was generated with the help of the ARTFL dictionary. First, the 500 most frequent words in the million-word Hansard subset (excluding punctuation) were extracted. These terms were then matched against the ARTFL dictionary, removing those words which had multi-word translations as well as several which listed multiple parts of speech for the same translation (multiple parts of speech can only be used if the corresponding translations are distinct from each other). The remaining 420 translation pairs, tagged for part of speech, were then converted into seed clusters and provided to the clustering program. To facilitate experiments using the pre-existing production-rule grammar, five additional translation pairs from the manually-generated equivalence classes were added to provide seeds for five equivalence classes which are not present in the dictionary.

5 Results

The method described in this paper does (subjectively) a very good job of clustering like words together, and using the clusters to generalize EBMT gives a considerable boost to the performance of the EBMT system.

Figure 2 shows a sampling of the smaller clusters generated from 1.1 million words of Hansard text. While the members of a cluster are often semantically linked (as in cluster 348, which contains types of political parties, or cluster 3539), they need not be. Those clusters whose members are not semantically linked generally contain words which are all the same part of speech, number, and gender (as in cluster 2472, which contains exclusively plural nouns) — but as will be discussed in the next section, even those clusters whose members are totally unrelated may be useful and correct. One fairly common occurrence among the smaller clusters is that various synonymous translations of a word (from either source or target language) will cluster together, as in cluster 1652. This is particularly useful when the target-language word is the same, as this allows various ways of expressing the same thing to be translated when any of those forms are present in the training corpus.

Figure 3 shows how adding automatically-generated equivalence classes substantially increases the coverage of the EBMT system. Alternatively, much less text is required to reach a specific level of coverage. The lowest curve in the graph is the percentage of the 45,000-word test text for which the EBMT system was able to generate translations when using strict lexical matching against the training corpus. The top-most curve shows the best performance previously achieved using both a large set of equivalence classes (in the form of tagged entries from the ARTFL dictionary) and a production-rule grammar (Brown, 1999). Of the two center curves, the lower is the performance when generalizing the training corpus using the equivalence classes which were automatically generated from that same text, and the upper shows the performance using clustering with the 425 seed pairs.

As can be seen in Figure 3, 80% coverage of the test text is achieved with less than 300,000 words using manually-created generalization information and with approximately 300,000 words when using automatically-created generalization information, but requires 1.2 million words when not using generalization. 90% coverage is reached with less than 500,000 words using manually-created information and should be reached with less than 1.2 million words using automatically-created generalization information, versus 7 million words without generalization. This reduction by a factor of four to five in the amount of text is accomplished with little or no degradation in the quality of the translations. Adding a small amount of knowledge in the form of 425 seed pairs reduces the required training text even further; this can largely be attributed to the merging of clusters which would otherwise have remained distinct, thus increasing the level of generalization.

Adding the production-rule grammar to the seeded clustering had little effect. When using more than 50,000 words of training text, the increase in coverage from adding the grammar was negligible, and even with the smallest training corpora the increase was very modest.

Using the same thresholds that were used in the fully-automatic case, clustering on 1.1 million words expands the initial 425 word pairs in 37 clusters to 3209 word pairs, and adds an additional 555 word pairs in 140 further non-trivial clusters. This compares very favorably

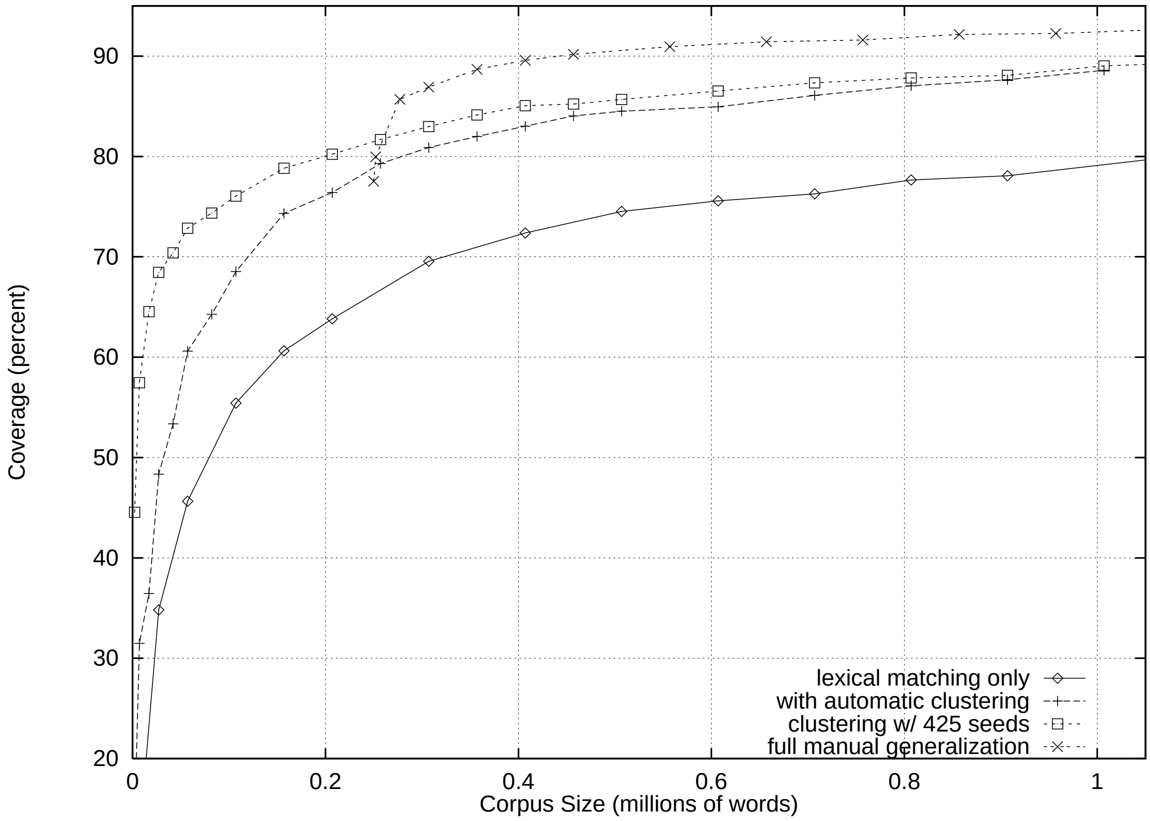

Figure 3: EBMT Performance with and without Generalization

to the 3506 word pairs in 221 clusters found without seeding.

The program also runs reasonably quickly. The step of creating context term vectors converts approximately 500,000 words of raw text per minute on a 300 MHz processor. For agglomerative clustering, the processing time is roughly quadratic in the number of word pairs, with a theoretical cubic worst case; the 17,527 distinct word pairs found from the million-word training corpus require about 25 minutes to cluster.

6 Discussion

One statement made earlier deserves clarification: the members of a cluster need not be related to each other in any way, either syntactically or semantically, for a cluster to be useful and correct. This is because (absent a grammar) we do not care about the features of the words in the cluster, only *whether their translations follow the same pattern.*

An illustration based on actual experience is useful here. In early testing of the group-average clustering algorithm with seeding, the `<conjunction>` seed class of “and” and “or” was used. Clustering augmented this seed class with “,” (comma), “in”, and “by”. One can easily see that the comma is a valid member of the class, since it takes the place of “and” in lists of items. But what about “in” and “by”, which are prepositions rather than conjunctions? If one considers the translation pattern

$$FreNP_1 \,\_\, FreNP_2 \;\rightarrow EngNP_1 \,\_\, EngNP_2$$

it becomes clear that *all* of the terms in the expanded class give a correct translation when placed in the blank in this pattern. Indeed, one could imagine a production-rule grammar geared toward taking advantage of such common translation patterns regardless of conventional linguistic features.

7 Conclusion and Future Work

Using word clustering to automatically generalize the example corpus of an EBMT system can provide the majority of the improvement which can be achieved using both a manually-generated set of equivalence classes and a production rule grammar. The use of a set of small initial equivalence classes produces a substantial further reduction in training text at a very low cost (a few hours) in labor.

An obvious extension to using seed clusters is to use the result of a clustering run as the initial seed for a second iteration of clustering, since the additional generalization of local contexts enabled by the larger seed clusters will permit additional expansion of the clusters. For such iterative clustering, all but the last round should presumably use stricter thresholds, to avoid adding too many irrelevant members to the clusters. Preliminary experiments have been inconclusive – although the result of a second iteration contains more terms in the clusters, EBMT performance does not seem to improve.

More sophisticated clustering algorithms such as k-means and deterministic annealing may provide better-quality clusters for better performance, at the expense of increased processing time.

This approach to generating equivalence classes should work just as well for phrases as for single words, simply by modifying the conversion step to create context vectors for phrases. This enhancement would eliminate the current limitation that translation pairs to be clustered must be single words in both languages. Work on this modification is currently under way.

An interesting future experiment would be foregoing grammar rules based on standard grammatical features such as part of speech, and instead creating a grammar guided by the clusters found fully automatically (without seeding) from the example text. The recent work by McTait and Trujillo (1999) on extracting translation patterns would appear to be a perfect complement, as they are in effect finding context strings with open slots, while the work described here finds the fillers for those slots. Given the ability to learn such a grammar without manual intervention, it would become possible to create an EBMT system using generalized examples from nothing more than parallel text, which for many language pairs could also be acquired almost fully automatically by crawling the World Wide Web (Resnik, 1998).

References

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

Sergio Barrachina and Juan Miguel Vilar. 1999. Bilingual Clustering Using Monolingual Algorithms. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 77–87, Chester, England, August.

Peter Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16:79–85.

Ralf D. Brown. 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)*, pages 111–118, Santa Fe, New Mexico, July. `http://-www.cs.cmu.edu/~ralf/papers.html`.

Ralf D. Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-Based Translation System. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 22–32, Chester, England, August. `http://www.cs.cmu.edu/~ralf/-papers.html`.

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

Kevin McTait and Arturo Trujillo. 1999. A Language-Neutral Sparse-Data Algorithm for Extracting Translation Patterns. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 98–108, Chester, England, August.

Philip Resnik. 1998. Parallel Strands: A Preliminary Investigation into Mining the Web for Bilingual Text. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas (AMTA-98)*, volume 1529 of *Lecture Notes in Artificial Intelligence*, pages 72–82, Langhorne, Pennsylvania, October. Springer.

Tony Veale and Andy Way. 1997. Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation. In *Proceedings of the NeMNLP'97, New Methods in Natural Langauge Processessing*, Sofia, Bulgaria, September. `http://-www.compapp.dcu.ie/~tonyv/papers/-gaijin.html`.

Ellen M. Voorhees. 1986. Implementing Agglomerative Hierarchical Clustering Algorithms for Use in Document Retrieval. *Information Processing and Management*, 22(6):465–476.