Transfer-Rule Induction for Example-Based Translation

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

Abstract

Previous work has shown that grammars and similar structure can be induced from unlabeled text (both monolingually and bilingually), and that the performance of an example-based machine translation (EBMT) system can be substantially enhanced by using clustering techniques to determine equivalence classes of individual words which can be used interchangeably, thus converting translation examples into templates. This paper describes the combination of these two approaches to further increase the coverage (or conversely, decrease the required training text) of an EBMT system. Preliminary results show that a reduction in required training text by a factor of twelve is possible for translation from French into English.

1 Introduction

Lexicalist Example-Based Machine Translation (EBMT) systems such as those of Veale and Way (1997) and the author (Brown, 1999) have the advantage that they require little or no additional knowledge beyond the parallel text forming the example base, but the disadvantage that the example base must be quite large to provide good coverage of unrestricted texts. Since parallel texts of the required size are often difficult or (for less-used languages) even impossible to obtain, there have been several efforts to reduce the data needs by generalizing the training examples into templates and then perform template matching (see Figure 1).

Three approaches to generalization have been used: manually-generated equivalence classes, automatically-extracted equivalence classes, and transfer-rule induction. The EBMT systems mentioned above both convert translation examples into templates using manually-created information, such as from a machine-readable dictionary with part-of-speech information, to replace words with tokens indicating the class of word which may occur in a particular location. More recently, the author added automatically-generated equivalence classes using word-level clustering (Brown, 2000) and Cicekli and Güvenir (2001) have implemented transfer-rule induction from parallel text.

This paper reports the results of combining the latter two approaches, using transfer-rule induction followed by word-level clustering to find not only single words but also transfer rules which can be combined into equivalence classes.

2 Transfer-Rule Induction

To induce a set of grammar rules from a parallel corpus, we make the same assumption used by Cicekli and Güvenir (2000; 2001) and van Zaanen (2000): when two sentence pairs in the corpus have some part in common but differ in some other part, the similar and dissimilar parts each correspond to some coherent constituent. Note that such a "constituent" need not be a traditional constituent as used by linguists, such as a noun phrase or prepositional phrase; for our purposes, it suffices that the groupings which are found can be used interchangeably.

Initially, the system only searches for pairs of training instances where the source-language halves show the pattern

$$S_1 \; D \; S_2$$

where S_1 and S_2 are the same in both instances (at most one of these may be the empty string) and D differs between the two training instances, but may contain common subsequences. The algorithm is outlined in Figure 2 and described in detail below. Naturally, such a simple pattern will not capture all interesting phenomena; future work will address more complex patterns.

A recursive method is used to find sets of training instances with common word sequences at beginning or end ("initial string" and "final string" or "prefix" and "suffix"). After sorting the sentence pairs by their source-language sentences, one can simply perform a linear scan of the collection for runs of training instances with at least I words in common. Each run found determines a subcorpus on which we can search for runs with at least $I+1$ words in common. At each level in the recursion, sorting the training instances as though the order of their words were reversed allows the same

Training Input	205 delegates met in London. 200 delegates met in Paris.
String Match	delegates met in
Template Match	<number> delegates met in <city>.

Figure 1: String Match vs. Template Match

1. read the corpus into memory, creating a rough bitext mapping for each bilingual sentence pair
2. sort the corpus alphabetically by source-language sentence
3. for each F, find all sequences of sentence pairs which share the same first F words in the source language
4. for each sequence, create a subcorpus and:
   (a) sort the subcorpus alphabetically by **reversed** source-language sentence
   (b) for each L, find all sequences of sentence pairs which share the same last L words in the source language
   (c) for each sequence, create another subcorpus and:
      i. perform a pairwise comparison between sentence pairs, adding the differences to a new equivalence class and to the corpus. The bitext map is used to discard those differences which do not appear to match between source- and target-language sentences.
      ii. if sufficiently long, add the common initial/final strings to the corpus
5. apply the learned rewriting rules to the corpus, except to sentence pairs where doing so would generate a single token
6. repeat steps 2 through 5 until no more new equivalence classes are added or the number of iterations reaches a preset maximum

Figure 2: The Induction Process

type of scan to find runs of training instances with a common final string of specified length.

Consider the small set of example sentence pairs in Figure 3. These all share the common initial string “nous regardons” and final string “.”; further, all but the first one share the initial string “nous regardons les”, and instances 2-4 share the initial string “nous regardons les approvisionnements en”. We will first process the smallest set (instances 2-4), then the intermediate set (instances 2-5), and finally the complete set (instances 1-5).

Thus, for each combination of initial-string length and final-string length, a set of training instances has been defined by the above scans, whose differences may be assigned to an equivalence class. These instances are then compared pair-wise to determine the differences between them. For each pair, the target-language halves are compared, also segmenting them into a common initial string, dissimilar central portion, and common final string. To ensure that the dissimilar center does in fact correspond to the difference on the source-language side, a bitext mapping is used. The bitext mapping is generated for each training instance from a bilingual dictionary, indicating which words in the target-language half potentially correspond with each source-language word. If, for either of the two training instances, the bitext map rules out any part of the target-language difference, neither instance is added to the current equivalence class (one or both of the instances may eventually have its center portion added when compared against other sentence pairs in the corpus).

Should a pair of instances pass the bitext-map test, the portions which differ between the two are added to the training corpus as new (but shorter) training instances, and are added to the equivalence class of all instances having the same initial and final strings. After the pair-wise comparison between each pair in the set of instances with that common prefix and suffix is complete, the initial and final strings themselves are also added to the corpus as two additional training instance provided that they are sufficiently long (currently, at least two words each).

Once the corpus has been completely processed, the result is a corpus augmented by various sentence fragments which are assumed to be constituents of some kind. We now apply the learned equivalence classes interpreted as a set of rewriting rules or a context free grammar, replacing each instance of a class member by the class name.

Input:

1. nous regardons la production de acier .
   we are watching steel production .
2. nous regardons les approvisionnements en énergie .
   we are watching energy supplies .
3. nous regardons les approvisionnements en engrais .
   we are watching fertilizer supplies .
4. nous regardons les approvisionnements en matériel .
   we are watching equipment supplies .
5. nous regardons les produits chimiques agricoles .
   we are watching agricultural chemicals .

Rewritten Corpus:

- nous regardons la production de acier .
  we are watching steel production .
- nous regardons les <cl_2> .
  we are watching <cl_2> .
- nous regardons
  we are watching

Induced Rules – <CL_2>:

- approvisionnements en <CL_0>
  <CL_0> supplies
- produits chimiques agricoles
  agricultural chemicals

Induced Rules – <CL_0>:

- engrais
  fertilizer
- matériel
  equipment
- énergie
  energy

Figure 3: Sample Induction

The replacements may occur anywhere in a sentence pair, including the portions which had been used as common prefix or suffix strings during the learning phase. An exception is made if the end result of applying the grammar to a training instance is a single class name; in this case, the training instance is left unchanged. The rationale for applying rewriting rules in this manner is to increase the similarity of the items in the corpus to permit more matches on the next iteration, e.g. two instances which previously had different initial segments may have the same initial segment after rewriting rules are applied, because the differing phrases are both members of the same equivalence class.

At this point, if any changes were made, the entire process repeats using the updated corpus, until no more equivalence classes can be created. To forestall an extremely lengthy execution time should a large number of iterations be required, the program can also terminate learning after a specified number of iterations.

After the induction completes, it can optionally be re-run in the reverse direction, comparing target-language sentences with each other. This feature can increase the yield of equivalence classes by 50% or more, since the target-language sentences will show different patterns of similarities with each other which were not captured during the first pass.

Figure 3 shows the result of this process. The pairwise comparison between instances 2 through 4 yields the single-word differences which have been added to equivalence class <CL_0>. The further comparisons between instances 2 through 5 yield the equivalences in <CL_2>; after applying the rewriting rules created by <CL_0>, three of its members collapse into a single rule containing an equivalence-class marker.

The output of the induction phase is a set of

1. find bilingual word pairs which uniquely correspond to each other according to a bitext mapping generated with a bilingual dictionary
2. accumulate counts for the words in the immediate vicinity of each occurrence of a word pair, as well as the frequency of the word pair itself
3. convert the word counts into weighted term vectors, and generate an initial cluster for each vector; associate the word pair's frequency with the term vector
4. repeatedly find the two most similar clusters whose similarity is above the threshold value for the lesser of their frequency values and merge them (adding the associated frequencies), until no pair of clusters is above threshold or only two clusters remain
5. if the number of clusters is still above 2500, relax the clustering by allowing those clusters with frequency values of 5 or less to merge regardless of the clustering threshold for their frequency (if necessary, repeat for frequencies of 10, 15, etc.)
6. output the clusters with more than one member, recovering the word pair with which each vector in a cluster is associated

Figure 4: Clustering Algorithm

parallel rewriting rules which form the transfer-rule grammar (see Figure 5 for a few examples from actual runs), and optionally an updated parallel corpus with the rewriting rules added and already applied. The updated corpus which is already present in the computer's memory can then be used as input to the word-clustering phase.

3 Word Clustering

To cluster words into equivalence classes, we used the approach of (Brown, 2000), outlined in Figure 4. The main feature of this approach is a transformation step which converts the word-clustering problem into a document-clustering problem.

The first step in word clustering is to determine *which* words should be clustered. Since it is also necessary to cluster bilingually, a dictionary is used to generate a rough bitext mapping between the source and target halves of each sentence pair in the training corpus. Whenever the bitext map indicates a unique correspondence between a word in the source-language sentence and some word in the target-language sentence, form a word pair from the source- and target-language words and treat it as an indivisible unit. These word pairs are what will be clustered.

For each occurrence of a word pair, add the source-language words immediately surrounding its occurrence (for these experiments, the three words preceding and the three words following) to a term vector which tallies all neighboring words across all occurrences of the word pair. This converts the problem into one of finding which term vectors cluster together, a standard document-clustering approach.

Next, the term vectors are clustered using bottom-up agglomerative clustering. Initially, one cluster is created for each vector; next, the two clusters with the highest similarity measure are merged, and the process is repeated until no more clusters have sufficiently high similarity with any other clusters. The similarity metric used is a term-weighted cosine similarity measure, e.g. the normalized inner product of the two vectors:

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}|| \times ||\vec{v}||}$$

The selected threshold for clustering proved to be excessively conservative for larger training sets, leaving a majority of word pairs in single-element clusters, so a back-off scheme was added to force the total number of final clusters closer to a target of 2500 clusters. If more than the target number of clusters remain when the first clustering pass terminates, clusters with frequency values (the sum of term frequencies for all word pairs included in the cluster) of five or less may merge regardless of the selected threshold. If necessary, further iterations will allow clusters with frequencies of up to 10, 15, 20, etc. to merge. This approach was selected to avoid the need for tuning the thresholds for each individual training set size, which would be impractical at best.

Once the vectors have been clustered, the clusters are output. For every cluster, the word pair to which each vector in the cluster corresponds is recovered and is written to the results file along with the name of the cluster. The results file becomes input to the EBMT system when it indexes the training corpus, allowing it to convert the parallel text into templates using the cluster names as class markers. Figure 6 shows a sample of the clusters produced from 107,000 words of parallel training text. The clustering process produced a total of 1178 clusters containing 3187 word pairs after discarding clusters containing only a single word pair (as such singletons are not useful).

<CL_18>:
plutôt ridicule
rather ridiculous

peut-être le condamner avec de fausses louanges , mais je ne y puis rien
If it is damning him with feint praise , I cannot help

<cl_847> monsieur l'Orateur
<cl_847> Mr . Speaker

bien pour Toronto
lucky in Toronto

...

<CL_119>:
question # 2 est oui
part two is yes

question # 2601 de la première session de la 29e législature ne tenait pas compte de les travaux de construction entrepris en 1973 - 1974 à la résidence de le premier ministre à Ottawa
question No . 2,601 of the First Session of the twenty-ninth Parliament did not list construction work carried out in the year 1973 - 74 at the Prime Minister's Ottawa residence

<CL_122>:
, confié à la garde de un service de sécurité , fut transporté par tout le pays à bord de un avion de la Défense nationale et un exemplaire en a été remis à toutes les capitales provinciales au moment même où je prenais la parole à la Chambre
was sent under security guard in national defence aircraft right across the country , and copies were released in all provincial capitals simultaneously with my rising in the House

que <cl_644>
that the <cl_644>

Figure 5: Sample Replacement Rules

4 Combining Induction and Clustering

The induction process produces a large number of usually small equivalence classes, which limits the amount of generalization that can be produced. Hence, we would like to merge different classes of rewriting rules which are used in similar contexts, in the same manner as individual words are clustered.

A by-product of the induction algorithm is an updated corpus with all replacement rules already applied, leaving single-word markers in place of the phrases found by transfer-rule induction. Applying the clustering process to the modified corpus allows both words and replacement rules to cluster together. While replacement-rule nonterminals tend to cluster with other nonterminals, many clusters contain both words and nonterminals (see Figure 6).

5 Experimental Design

To gauge the effect of the different approaches, various combinations of method and training size were run and evaluated.

Four conditions were compared: simple string matching against the corpus (see Figure 1), matching templates formed using single-word clustering alone, templates formed using transfer-rule induction alone, and templates formed with the combination of clustering and induction. For French-English, the training data in each case consisted of a subset (up to 1.1 million words in 19730 sentence pairs) of the Hansard corpus made available by the Linguistic Data Consortium (Linguistic Data Consortium, 1997) and a bilingual dictionary formed by combining the ARTFL French-English dictionary (ARTFL Project, 1998) with a probabilistic dictionary extracted from the Hansard corpus. The test data consisted of 45,320 words of French text from a disjoint portion of the Hansard corpus.

The data for the Spanish-English experiments consisted of up to one million words of parallel text drawn primarily from the UN Multilingual Corpus(Graff and Finch, 1994) available from the Linguistic Data Consortium and a bilingual dictionary derived from the Collins Spanish-English

Cluster	French	English
507	NE	NOT
	NE	NO
524	SONT	ARE
	FURENT	WERE
568	CETTE	THIS
	CETTE	THAT
575	PROCHAIN	NEXT
	DERNIER	LAST
659	UNE	THE
	UNE	AN
	UNE	A
	UN	THE
	UN	A
	LE	THE
	LE	OF
	LE	IT
	LE	IN
	LE	A
	LA	THE
	LA	OF
	LA	IN
678	VALEUR	VALUE
	SÉCURITÉ	SECURITY
	PRODUCTION	PRODUCTION
	PLUS	MORE
	NÉCESSITÉ	NEED
	LIVRAISON	DELIVERY
	FA CON	WAY
	DATE	DATE
	CROISSANCE	GROWTH
	CONSTRUCTION	CONSTRUCT- ION
	COMMERCIAL- ISATION	MARKETING
1085	POINTS	POINTS
	LIVRES	POUNDS
	ANS	YEARS
1150	VALABLES	VALID
	TARD	LATER
1196	MILLIONS	MILLION
	MILLIARDS	BILLION
1776	ABSURDE	NONSENSE
	<CL_18>	<CL_18>
2158	PÊCHEURS	FISHERIES
	PÉNURIES	SHORTAGES
	OFFICIERS	OFFICERS
2609	<CL_54>	<CL_54>
	<CL_98>	<CL_98>
	<CL_375>	<CL_375>
	<CL_458>	<CL_458>
	<CL_462>	<CL_462>
	...6 more...	

Figure 6: Sample Clusters from 107,000 words

dictionary and statistically extracted from UN-corpus and other parallel text. The test data consisted of 9,059 words of Spanish test from a disjoint portion of the UN Multilingual Corpus.

The input to the actual EBMT system consisted of the bilingual dictionary (for word-level alignment) plus the original parallel text, and (as appropriate) the output of the clustering and/or induction phases. When the clustering algorithm was applied in isolation, the single-word rewriting rules found through clustering were supplied to EBMT. When the transfer-rule induction was used, the induced transfer rules were supplied to EBMT.

The transfer-rule induction was limited to twelve iterations in each direction. During development it was found that the process has largely converged after six iterations, even though total convergence may require fifteen or more iterations, with the last several iterations each adding only a few new equivalence classes with a handfull of members.

The performance measure used to determine the effectiveness of the various methods is the coverage of the test text, i.e. the percentage of the total words in the test input for which the EBMT system could generate at least one candidate translation. Although this metric does not measure quality, the design of the EBMT system generally enforces some minimum level of translation quality – the translation software will not output translations when the word-level alignment for the retrieved training example fails or is deemed too poor[1]. Recent manual judgements on a Mandarin Chinese-English version of the EBMT system (Zhang et al., 2001) have confirmed that increased coverage indeed correlates with improved translation quality.

Figure 7 shows some sample output, which will be discussed in more detail in Section 7; for the moment, it is important to note that the score shown is a penalty – 0.0 is considered a perfect alignment, while matches for which the penalty exceeds five times the number of words are not output at all.

6 Computational Complexity

Each iteration of the induction algorithm takes time $O(n^2)$, where n is the number of words of training text, since ultimately each sentence pair must be compared against every other sentence pair (subdividing the problem into shorter sequences does not increase the time complexity, and sorting is $O(n\ log\ n)$). The number of iterations required to run the induction to completion

[1] In fact, if all corpus matches yielded good alignments, coverage would be in excess of 90% instead of 80.14% for two million words of French-English training text using just string matching.

Il me semble qu'il conviendrait maintenant de reprendre le débat.
I think it would be proper at this time for the debate to continue.

String Match, 1.1 million words:

Match	Sc	Translation
il me semble qu'il	0	it seems to me that
il me semble que	0	It seems to me that
il me semble	0	It seems to me
il me	0	Let me
il me	0	me explore
me semble	0.3	seems to me
me semble	0	I think
semble que il	0	it seems this
semble que	0	seems that
que il	0	well as
maintenant de	2.5	now the
de reprendre	0	to resume
le débat .	1	in debate .
le débat .	0	the debate .
le débat	0	in debate
le débat	0	debate in
débat .	0	debate .

Induction+Clustering, 1.1 million words:

Match	Sc	Translation
il me semble qu'il	17	it seems to me that
semble qu'il	0	it seems this
qu'il	0	that he
il conviendrait	0	IT APPROPRIATE
maintenant	0	NOW
de reprendre	0	to resume
le débat .	0	THE DEBATE .
le débat .	1	in DEBATE .
débat .	0	DEBATE .

Figure 7: Sample Translation 1

is potentially $O(n)$, but appears in practice to be somewhat less than $O(log\ n)$; there is a three- to four-fold increase between a 50,000-word corpus and a twenty times larger million-word corpus. More experimentation with larger corpora (several to tens of millions of words) will be required to determine the actual value.

In practice, the first iteration takes the longest time, by a factor of two or more, and subsequent iterations complete more quickly. Per-iteration execution times generally continue to decrease until the fifth iteration, after which they tend to vary both up and down but stay relatively constant. Two factors are likely at work here: on each succeeding iteration, there are fewer and smaller runs of sentences, reducing the quadratic pair-wise comparison; and the individual training instances are shorter, either because they are fragments of an older instance or due to replacement of phrases by single tokens.

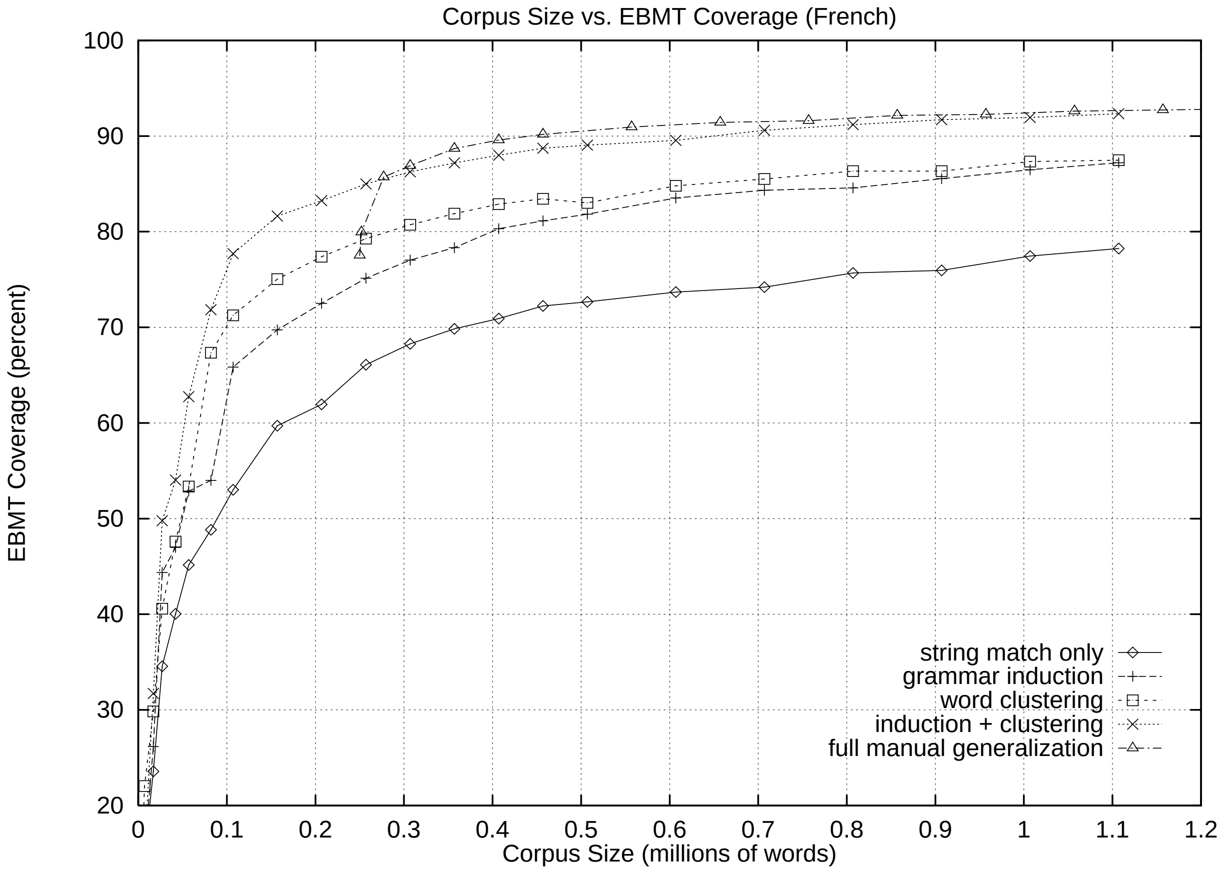
The clustering algorithm is also $O(n^2)$, but here n is the number of term vectors, i.e. the number of distinct bilingual word pairs, which grows more slowly than the number of words in the training text. Thus, the execution time for the complete process of induction plus clustering is slightly worse than quadratic in the size of the input.

7 Results

As shown in Figure 8, clustering and transfer-rule induction each outperformed simple string matching, and the combination substantially outperformed both. In fact, the combined algorithm exceeds the coverage of string matching trained on two million words of French-English parallel text with only 157,000 words of training data, more than a twelve-fold reduction in training data with no additional knowledge sources. For comparison, the best results (Brown, 1999) achieved using manually-created generalization information consisting of a large part-of-speech tagged bilingual dictionary and several hundred bilingual production rules based on those tags are shown in the graph as well. The automatic algorithms very nearly match the performance of the manual approach, without the quarter-million words of additional data in the dictionary and grammar rules used by manual generalization.

Similar results were obtained for Spanish-English (see Figure 9), where the combined algorithm had greater coverage with 104,000 words of training data than string matching on a one-million-word corpus.

Initial evaluation of the translation quality showed that most of the degradation in quality from grammar induction or the combination of grammar induction and clustering was due to misalignments, where the translation found by the system included extraneous words or omitted a portion of the true translation. The most egregious case was fairly easy to avert, simply by not applying rewriting rules to a new sentence pair if, after applying the rules, it has the form "<equivclass>" == "<equivclass> extra words" (or vice-versa). While the extra word(s) might be appropriate for the particular phrase, it is unlikely that they will be appropriate for all members of the equivalence class. This limitation reduced the coverage slightly, but substantially improved the quality of the EBMT system's output. A stricter consistency check than the current test of whether the bitext mapping positively rules out any part of the candidate translation would most likely further improve the translation quality, although the current system shows only minor degradation. Much of that degradation can be attributed to overgeneralization to cases where the



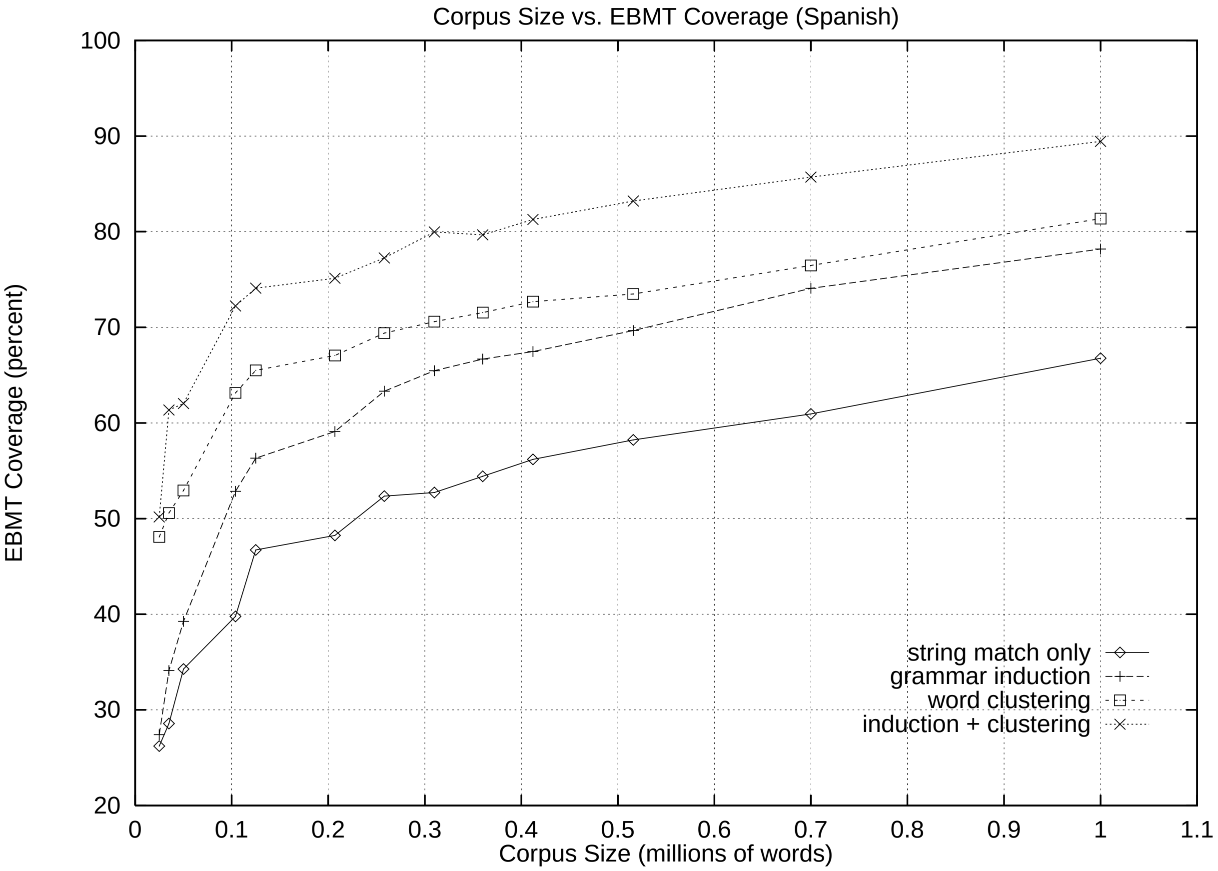
Corpus Size (words)	String Match Only	Induction	Clusters	Clust+Induc
107,000	53.01%	65.85%	71.25%	77.70%
157,000	59.72%	69.73%	75.05%	81.64%
207,000	61.94%	72.51%	77.38%	83.25%
307,000	68.28%	77.02%	80.73%	86.27%
1,107,000	75.26%	87.20%	87.33%	92.34%
2,007,000	80.14%	–	–	–

Figure 8: Performance Comparison (French)

usual default translation rule does not apply.

Some examples of the EBMT output are shown in Figures 7 and 11; the reference translation from the Hansard corpus is given for each. Figure 11 includes a very terse form of the output due to the limited space available; all matches which are contained within some longer match (from another example in the corpus) have been excluded, and only the best-scoring match from among those covering a particular phrase is shown. The actual output is several times longer, and includes some translations with better quality than the maximal matches actually shown. For each match, the alignment score (lower is better) and the translation are shown. Words in all-capital letters indicate where a single word matched an equivalence class generated by clustering, rather than the surface string. The sentence in Figure 7 was randomly selected from among the shorter sentences in the test set, while Figure 11 is the very last sentence in the test set.

Shortly before the final version of this paper was submitted, some changes were made to the EBMT system to improve its run-time efficiency. As part of those changes, some tweaks were made to the word-level alignment algorithm, including the generation of the correspondence table used for clustering as well as word-level alignment during translation. Those tweaks have resulted in somewhat paradoxical and as yet unexplained changes in the system's coverage (see Figure 10) – coverage for string matching and grammar induction dropped considerably, while clustering is greatly improved and the combination of induction and



Corpus Size (words)	String Match Only	Induction	Clusters	Clust+Induc
104,000	39.78%	52.86%	63.15%	72.23%
207,000	48.25%	59.11%	67.05%	75.14%
310,000	52.72%	67.47%	70.60%	79.98%
1,000,000	66.77%	78.19%	81.36%	89.44%

Figure 9: Performance Comparison (Spanish)

clustering remains almost unchanged[2]. Now that performance of clustering alone is so much closer to the performance of the combined algorithm, it becomes clear that the changes in the corpus produced by grammar induction interfere somewhat with clustering. With small training sets, the combined algorithm actually fares somewhat worse than clustering alone.

8 Conclusion

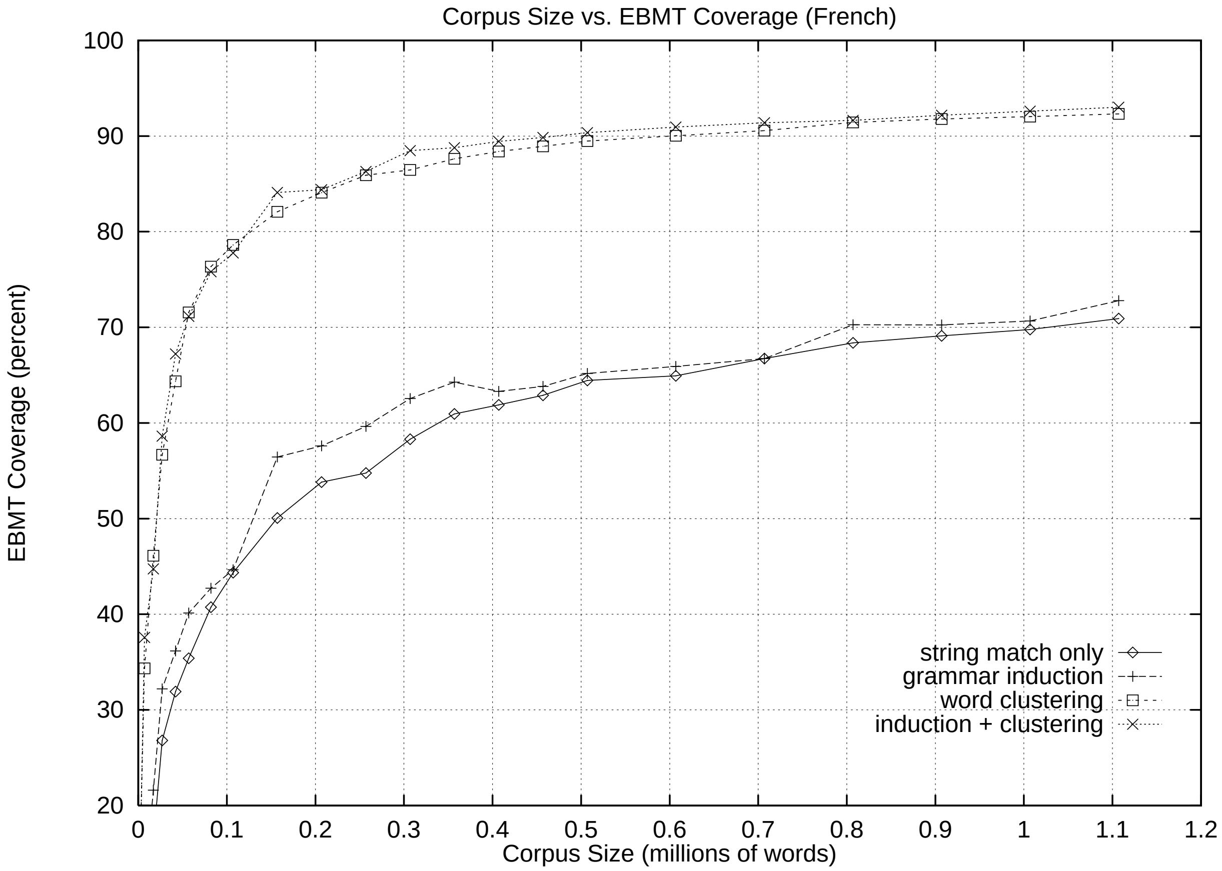
Experimental results indicate that combining transfer-rule induction in the style of (Cicekli, 2000) with the author's prior work on single-word clustering is beneficial, resulting in a system that outperforms either method used in isolation and dramatically reducing the amount of parallel training text required for a broad-coverage EBMT system. Coverage for a given amount of training text is increased with little or no impact on translation quality.

[2] The restriction imposed on grammar induction to avoid bad translations slightly reduced the performance, and then the changes to the EBMT system somewhat improved coverage.

9 Ongoing and Future Work

The applicability of this approach has already been shown for two language pairs, but its effectiveness for very divergent language pairs remains to be demonstrated. Future experiments will include Mandarin-English as well as French-English and Spanish-English.

The transfer-rule induction can certainly be enhanced, for example by checking for common sequences within the dissimilar center portions, allowing them to be split even further. Currently, the learned equivalences tend to be fairly long; shorter phrases will be more general and more



Corpus Size (words)	String Match Only	Clusters	Induction	Clust+Induc
107,000	44.35%	78.62%	44.66%	77.77%
207,000	53.81%	84.09%	57.61%	84.38%
307,000	58.31%	86.46%	62.56%	88.48%
1,107,000	70.92%	92.32%	72.79%	93.01%

Figure 10: Updated French Performance

likely to be matched in previously-unseen text during translation (improving coverage).

The word clustering parameters need to be tuned. Currently, the same parameters are used in conjunction with transfer-rule induction and in isolation. There is no *a priori* reason for the optimal settings in one case to be optimal for the other. In addition, the target of 2500 clusters was chosen arbitrarily and should be tuned for the best trade-off between quality and coverage.

The interference between transfer-rule induction and clustering noted during the most recent experimental runs should be isolated and, if possible, mitigated.

Finally, there is the possibility that adding a small amount of seed knowledge to the grammar induction process (similar to what has was previously done with single-word clustering) could substantially improve performance. Such seeding will require additional support in the software.

10 Acknowledgements

This research was supported primarily by the National Science Foundation's STIMULATE program (grant number 9618941). The author would like to thank the anonymous reviewers for their helpful comments, which have considerably improved the quality of this paper.

References

ARTFL Project. 1998. *ARTFL Project: 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`.

Ralf D. Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-Based Translation

Comme il est 2 heures 03, la Chambre s'ajourne à 11 heures aujourd'hui, conformément au paragraphe 24(1) du Règlement.
It being 2.03 a.m., this House stands adjourned until later this day at ll a.m., pursuant to Standing Order 24(1).

String-Match Only, 1.1 million words:

Match	Score	Translation
comme il	1	He just a couple of
il est 2	1.275	it is 2
est 2 heures	1.275	is 2 o'clock
03 , la chambre	1	03 , the House
, la chambre se	1	, the House will
à 11 heures	0	at 11 a.m.
hui , conformément	1	today , pursuant
paragraphe 24 (1) de le règlement .	41.15	subsection 24 (1) of the rule .

Induction+Clustering, 1.1 million words:

Match	Score	Translation
comme	0	LIKE
il est 2 heures	1.825	IT IS 2 o'clock
03 , la chambre	0.75	03 , THE HOUSE
à 11 heures	1.11	to 11 a.m.
hui , conformément	1	today , pursuant
conformément au paragraphe 24 (1) de	2.65	ACCORDANCE TO SUBSECTION 24 (1) of
paragraphe 24 (1) de le règlement .	1.55	subsection 24 (1) OF STANDING .

Figure 11: Sample Translation 2

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`.

Ralf D. Brown. 2000. Automated Generalization of Translation Examples. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000)*, pages 125–131.

Ilyas Cicekli and H. Altay Guvenir. 2001. Learning Translation Templates from Bilingual Translation Examples. *Applied Intelligence.* (to appear).

Ilyas Cicekli. 2000. Similarities and Differences. In *Proceedings of SCI2000*, pages 331–337, July. `http://www.cs.bilkent.edu.tr/-~ilyas/pubs.html`.

D. Graff and 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, December. `http://-www.ldc.upenn.edu/`.

Menno van Zaanen. 2000. ABL: Alignment-Based Learning. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000)*, pages 961–967. `http://www.comp.leeds.ac.uk/menno/-docs/`.

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 Language Processessing*, Sofia, Bulgaria, September. `http://-www.compapp.dcu.ie/~tonyv/papers/-gaijin.html`.

Ying Zhang, Ralf D. Brown, and Robert E. Frederking. 2001. Adapting an Example-Based Translation System to Chinese. In *Proceedings of the Human Language Technology Conference 2001.* (to appear) `http://www.hlt2001.org/`.