Symmetric Probabilistic Alignment for Example-Based Translation

Jae Dong Kim, Ralf D. Brown, Peter J. Jansen, Jaime G. Carbonell

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{jdkim,ralf,pjj,jgc}@cs.cmu.edu

Abstract. Since subsentential alignment is critically important to the translation quality of an Example-Based Machine Translation (EBMT) system which operates by finding and combining phrase-level matches against the training examples, we recently decided to develop a new alignment algorithm for the purpose of improving the EBMT system's performance. Unlike most algorithms in the literature, this new Symmetric Probabilistic Alignment (SPA) algorithm treats the source and target languages in a symmetric fashion. In this paper, we describe our basic algorithm and some extensions for using context and positional information, compare its alignment accuracy with IBM Model 4, and report on experiments in which either IBM Model 4 or SPA alignments are substituted for the aligner currently built into the EBMT system. Both Model 4 and SPA are significantly better than the internal aligner and SPA slightly outperforms Model 4 despite being handicapped by incomplete integration with EBMT.

1. Introduction

The example-based translation system (Brown, 2000; Brown et al., 2003) used for the experiments described in this paper is internally a multiengine system (Frederking et al., 1994) consisting of a phrasal EBMT engine and word-for-word dictionary lookup which are combined using a language model-driven lattice search equivalent to the decoder used in many statistical machine translation systems[1]. The phrasal EBMT engine finds all partial matches between input sentences and the bilingual training corpus and then uses sub-sentential alignment between the halves of the retrieved examples to determine the corresponding translations. This makes the alignment process a critical factor in the quality of the system's translations.

Incorrect alignments result in missing or extraneous words in the translation hypothesized for a particular fragment of the input. This can cause critical information to be omitted, irrelevant words to be "hallucinated", or even invert the meaning of the text. Because retrieved training examples with low confidence alignments are not used, lack of a reliable alignment can result in the best translation for a given fragment being missed, or even no translation at all for that fragment of input. The latter is in fact a significant issue in the current implementation of EBMT, as it is quite common to have phrasal matches for 90% of the input words yet get translations for only 75% of the input words – the remaining matches did not produce any alignments that were considered satisfactory.

Because we recognized that the existing alignment algorithm is a substantial bottleneck in the system's translation performance, we decided to investigate a new algorithm and its effect on translation quality. In the remainder of this paper, we describe the existing aligner, the new SPA method, an initial evaluation of its alignment quality, how we incorporated the method into the EBMT engine, and a task-based evaluation of the resulting overall system using

[1] Other engines can be added to the system, but were not for these experiments, and are generally not available anyway.

the BLEU (Papineni et al., 2002) translation-quality metric.

2. Background

The existing alignment algorithm used by the EBMT engine is a very fast but heuristic algorithm based around a correspondence table created from bilingual lexicon lookups (Brown, 1997) with some additional processing to reduce ambiguity and fill gaps. When aligning a phrasal match, the correspondence table is consulted to find words with unique correspondences as anchor points, and then the resulting match is extended on each side until a target language word is reached which could not be a translation of any word in the source match. All contiguous sub-phrases of this largest possible match, which also include all anchor points are scored based on a weighted sum of various heuristic functions such as percentage of words with known correspondences and difference in length between source and target, and the highest-scoring sub-phrase is output as the candidate translation.

The primary advantage of this method is its great speed. By pre-computing the correspondence table and storing it in the indexed corpus, the system can perform upwards of 20,000 phrasal alignments per second on a 2 GHz PC. Computing the correspondence table is more expensive but, even though it is the most computationally expensive portion of the training, still permits indexing of over 1000 sentence pairs per second. The main drawback is that alignment decisions are based on binary correspond/ don't correspond decisions rather than using translation probabilities to decide between ambiguous alternatives. This makes the dictionary of critical importance to the quality of the alignments, especially in its selection of cut-offs for including translations – low-probability translations are treated the same as high-probability translations by the correspondence table. Further, while one or two gaps in dictionary coverage for a particular sentence can be handled, insufficient coverage becomes a large problem as the size of the training data decreases when there is no external bilingual lexicon available.

In contrast to this very simple aligner, the IBM Model 4 algorithm computes translation probabilities, distortion probabilities, and word classes for reordering, and uses them to determine the globally most probable alignments of the words in a sentence pair. We selected Model 4 to compare against our new algorithm because it is widely used and understood, and an implementation was readily available in the GIZA++ program in the EGYPT toolkit (Al-Onaizan et al., 1999).

3. Symmetric Probabilistic Alignment (SPA)

In sub-sentential alignment, mappings are produced from words or phrases in the source language sentence and those words or phrases in the target language sentence that best express their meaning.

An alignment algorithm takes as input a bilingual corpus consisting of corresponding sentence pairs and strives to find the best possible alignment in the second for selected n-grams (sequences of n words) in the first language. The alignments are based on a number of factors, including a bilingual dictionary (preferably a probabilistic one), the position of the words, punctuation, invariants (such as numbers), and so forth.

For our baseline algorithm, we make the following simplifying assumptions, each of which we intend to relax in future work:

1. A fixed bilingual probabilistic dictionary is available.
2. Contiguous fragments (word sequences) of source language text are translated into contiguous fragments in the target language text.
3. Fragments are translated independently of surrounding context.

3.1. A Baseline Algorithm

Our baseline algorithm is based on maximizing the probability of bi-directional translations of individual words between a selected n-gram in the source language and every possible n-gram in the corresponding paired target language sentence. The reason why we use the probability of bi-directional translations is that we are more convinced when both side's fragments agree that the other side's fragment is its translation. For example, given a source fragment S_{F_i}, let's say two target fragments T_{F_k} and T_{F_l} are equally most probable to be S_{F_i}'s translation. If

we consider opposite directional translations and find that T_{F_k} 's the most probable translation is S_{F_i} and T_{F_l} 's the most probable translation is S_{F_j} $(i \neq j)$, we will choose T_{F_k} as the translation of S_{F_i}.

No positional preference assumptions are made, nor are any length preservation assumptions made. That is, an n-gram may translate to an m-gram, for any values of n or m bounded by the source and target sentence lengths, respectively. Finally a smoothing factor is used to avoid singularities (i.e. avoiding zero-probabilities for unknown words, or words never translated before in a way consistent with the dictionary).

Suppose that we are given a pair of aligned sentences *S* and *T* where a source sentence S is

$$S : w_0, w_1, ..., w_i, ..., w_{i+k}, ..., w_n \quad (1)$$

and the corresponding target language sentence *T* is

$$T : v_0, v_1, ..., v_j, ..., v_{j+l}, ..., v_m \quad (2)$$

and calculating the translation probabilities between a source fragment S_F and target fragments in $\{T_F\}$.

Then the segment we try to obtain is the target fragment $\overline{F_T}$ with the highest probability of all possible fragments of S2 to be a mutual translation with the given source fragment, or

$$\overline{F_T} = \arg\max_{\{F_T\}} (p(w_i, ..., w_{i+k} \leftrightarrow v_j, ..., v_{j+l})) \quad (3)$$

All possible segments can be checked in O(m^2), where m is the target language length, because we will check *m* 1-word-length segments, *m-1* 2-wordlength segments, and so on. If we bound the target language n-grams to a maximal length *k*, then the complexity is linear, i.e. *km*.

We compute the score of the best possible alignment as follows:

Given a source fragment $w_i, ..., w_{i+k}$, a target fragment $v_j, ..., v_{j+l}$, let's say $\{w_i, ..., w_{i+k}\}$ is the set of the source fragment words and $\{v_j, ..., v_{j+l}\}$ is the set of the target fragment words. For a source word $w_{i+a} (0 \le a \le k)$, we define the *translation relation probability* as follows:

$$p(Tr(w_{i+a}) \in \{v_j, ..., v_{j+l}\}) = \max(p(v_{i+b} \mid w_{i+a}),\ 0 \le b \le l \quad (4)$$

Then the score of the best alignment is

$$Score_{\hat{F}_T} = \max_{\{F_T\}} Score_{F_T} \quad (5)$$

where the score can be written as two bi-directional components

$$Score_{F_T} = P1 \times P2 \quad (6)$$

where *P1* is the product of the translation probabilities of the source fragment words and *P2* is the product of the translation probabilities of the target fragment words.

These can be further specified as

$$P1 = \left(\prod_{m=0}^{k} \max(p(Tr(w_{i+m}) \in \{v_j, ..., v_{j+l}\}), \varepsilon) \right)^{\frac{1}{k+1}} \quad (7)$$

$$P2 = \left(\prod_{n=0}^{l} \max(p(Tr(v_{j+n}) \in \{w_i, ..., w_{i+k}\}), \varepsilon) \right)^{\frac{1}{l+1}} \quad (8)$$

where ε is a very small probability used as a *smoothing value*.

3.2. Length Penalty

The ratio of target segment (n-gram) and source segment (m-gram) lengths should be comparable to the length ratio of the target sentence and source sentence lengths, though certainly variation is possible. Therefore, we generate a penalty function to the alignment probability that increases with the discrepancy between the ratios as *n/m* is compared to the source/target sentence length ratio.

Let the length of the source language segment be *i* and the length of a target language segment under consideration be *j*. And let the dynamic sentence length ratio be $R_{|T|/|S|}$ given a source language sentence *S* and its corresponding target language sentence *T* in the corpus. The *expected target segment length* is then given by $\hat{j} = i \times R_{|T|/|S|}$. Further defining an *allowable difference AD*, our implementation calculates the length penalty LP_{F_T} as follows:

$$LP_{F_T} = \min\left(\left(\frac{|j - \hat{j}|}{AD} \right)^4, 1 \right) \quad (9)$$

We wanted to ignore target candidate fragments which have larger difference than *AD* and to give bigger penalty to the AD-satisfying target candidate fragments as they have larger difference. For equation (9), we tried powers of 2 through 6 and 4 gives the best results in our experiments.

The score for a segment including the penalty function is then:

$$Score_{F_T} \leftarrow Score_{F_T} \times (1 - LP_{F_T}) \qquad (10)$$

Note that, as intended, the score is forced to 0 when the length difference $| j - \hat{j} | > AD$.

3.3. Distance Penalty

Closely related languages (such as French and English) tend to have more similar word orders than more distantly-related languages such as Korean and English. In the former case, this results in greater phrase order similarity and consequently similar phrase positions.

In such a close language pair, we introduce a distance penalty to penalize the alignment score of any candidate target fragment which is out of the expected position range. First, we calculate C_E, the expected center of the candidate target fragment using C_{F_S}, the center of the source fragment and the dynamic sentence length ratio $R_{|T|/|S|}$.

$$C_E = C_{F_S} \times R_{|T|/|S|} \qquad (11)$$

Then we calculate an allowed distance limit of the center $D_{allowed}$ using a constant limit value $CON_{distance}$ and the dynamic sentence length ratio $R_{|T|/|T|_{average}}$ where $|T|_{average}$ is the average target sentence length in the training corpus.

$$D_{allowed} = CON_{distance} \times R_{|T|/|T|_{average}} \qquad (12)$$

Let D_{actual} be the actual distance difference between the candidate target fragment's center and the expected center, and set

$$Score_{F_T} \leftarrow \begin{cases} Score_{F_T}, \text{ if } D_{actual} \geq D_{allowed} \\ \dfrac{Score_{F_T}}{(D_{actual} - D_{allowed} + 1)^2}, \text{ otherwise} \end{cases} \qquad (13)$$

We also wanted to ignore target candidate fragments which have longer distance than the allowed distance. Equation (13) is derived empirically under the idea of giving bigger penalty to more distant target candidate fragments.

Furthermore, we think that we can apply this penalty to language pairs which have lower word order similarities. Because there might exist certain position relationship between such language pairs, if we can calculate the expected position using each language's sentence structure, we can apply a distance penalty to them.

3.4. Anchor Context

If the adjacent words of the source fragment and the candidate target fragment are translations of each other, we expect that this alignment is more likely to be correct. We boost $Score_{F_T}$ with the anchor context alignment score $Score_{AC_p}$,

$$Score_{AC_p} = P(w_{i-1} \Leftrightarrow v_{j-1}) \times P(w_{i+k} \Leftrightarrow v_{j+l}) \qquad (14)$$

$$Score_{F_T} \leftarrow (Score_{F_T})^{\lambda} \times (Score_{AC_p})^{1-\lambda} \qquad (15)$$

Empirically, we found this combination gives the best score when $\lambda = 0.7$ and it gave a better result than

$$Score_{F_T} \leftarrow \lambda \times Score_{FT} + (1 - \lambda) \times Score_{AC_p} \qquad (16)$$

4. Experimental Design

We set up two kinds of experiments. One is to measure alignment accuracy and the other is to see whether our alignment method actually improves an EBMT system's performance.

For the first evaluation, we tested our alignment method on a set of English-French sentences (taken from the Canadian Hansard corpus) and on a set of English-Chinese sentences. In both cases, we compared the results of our algorithm to human alignments. Although the latter may not be perfect and sometimes are non-unique, they provide the only answer key available for repeatable tests. We report here the results in the more challenging of these two test sets, English-Chinese, where word order differences and sentence-length differences are most evident. As metrics, we use *precision*, *recall* and F_1 (the harmonic mean of precision and recall).

Let us suppose that our answer segment is $w_1, w_2, ..., w_k$ and the correct answer (human) segment is $hw_1, hw_2, ..., hw_l$. Note that correct (human) answer may be non-contiguous, but the combination of SPA and EBMT to date is only capable of using the best *contiguous* target *m*-gram alignment it can find. We compute the recall *R* and precision *P* as follows:

$$R = \frac{count(hw_i \in \{wj\})}{l} \quad (17)$$

$$P = \frac{count(w_i \in \{hw_j\})}{k} \quad (18)$$

To obtain an average alignment score for evaluation, we

1. generated all the possible source language sentence fragments lengths 3 through 8 from a set of 10 test sentences,
2. aligned those fragments by means of our algorithm, and
3. calculated the metrics given above by comparison with the human-aligned answers.

To have a better intuition of the alignment results we obtain for a given language pair (and corpus), we introduce the following as baselines: "random result", "positional result", and "best result".

The "random result" is a randomly chosen target segment regardless of the source segment, constrained to be of a length corresponding to the source segment normalized by the length ratio of the source and target sentences.

The "positional result" is a target segment whose position in the target language most closely matches the position of the source segment. We calculate the target segment's start and end position using source segment's start, end position and the length ratio of source sentence and target sentence. In particular, let the source sentence be of length *n* and the target sentence of length *m*, we expect source position *i* to correspond to target position *j* where $j \cong i \times \frac{m}{n}$.

The "best result" is the best contiguous target segment extracted from human alignments. To get the best result, first, we get human alignments for the sentence pairs which will be used to evaluate our algorithm. Then we choose a segment which has the largest harmonic mean value among human alignment segments and whole segment. Notice that the human alignment may not be contiguous, therefore "best alignment" represents the best that our algorithm could possibly perform.

For the second evaluation, to minimize the initial investment of effort for the EBMT evaluation, we performed a partial integration of the SPA and EBMT modules rather than fully incorporating SPA into the EBMT engine. In this partial integration, SPA is used to annotate the training corpus with alignments (both phrasal and word-to-word), and the annotations in the corpus override the EBMT engine's internal aligner. Phrasal alignments are stored as-is, and whenever a partial match against the corpus is exactly equal to the source half of such an alignment and has a score above a specified threshold, the target half is output as the candidate translation. The word-to-word alignments are used to build a correspondence table (overriding the one which would have been built in the absence of alignment annotations) and that table is consulted as usual to perform alignments of matches for which there is no phrasal alignment from SPA available. One drawback of this arrangement compared to a full integration is that we are unable to take advantage of non-contiguous alignments (however, such alignments would also require modifications to the decoder which have yet to be implemented).

This yields the following training regimens for the alignment methods. To test the old algorithm, we

1. build a statistical dictionary from the corpus
2. index the training text using that dictionary

To test performance with IBM Model 4 alignments, we

1. train GIZA++ on the training text
2. annotate the training corpus using Model 4
3. index the annotated corpus

To test performance with SPA, we

1. use GIZA++ to build a dictionary from the training text
2. run the SPA aligner on the training text using that dictionary
3. index the annotated corpus generated by SPA

The differently-trained translation systems are then each evaluated on the test set using the NIST and BLEU metrics.

For our EBMT experiments we used a subset of the IBM Hansard corpus available from the Linguistic Data Consortium. This corpus is divided into files of 10,000 sentence pairs (with an occasional garbled or missing line which has been removed prior to our use), of which we used only files 000 through 099. The training data consisted of the first 20,000 sentence pairs – essentially files 000 and 001 – for EBMT and the first 700,000 English sentences for the language model. The development test ("devtest") set used for parameter tuning consisted of the first 100 sentences of file 040 and the evaluation test set consisted of ten segments of 100 sentences drawn from files 060 and 080. Segmenting the evaluation test set in this manner allowed us to perform statistical significance tests.

In addition to testing the full IBM Model 4 and SPA algorithms with both phrasal and word-level alignments, we also tested the performance when using each algorithm restricted to generating pure word-level alignments for creating the correspondence table used by EBMT's internal aligner (in lieu of having it generate that table itself from the bilingual lexicon it extracted from the training examples). The runs using the restriction to word-level alignments are identified as "SPA-W" and "IBM4-W", respectively.

5. Results and Conclusions

For comparing the alignment accuracy, we chose the positional alignment as the base line – as this is the highest baseline we can achieve – and the best alignment as the goal. Table 1 shows the best result obtained by each alignment method. As previously mentioned, "positional" is the baseline for comparing alignment performance while "best" is the best possible selection of contiguous fragments.

Method	Recall	Precision	F1
random	0.199473	0.231560	0.214323
positional	0.682276	0.704527	0.693223
best	0.980430	0.913729	0.945905
SPA	0.747958	0.701463	0.723912
IBM4	0.645233	0.745311	0.691670

Table 1: Chinese-English alignment accuracy results

After separately tuning several key parameters in the EBMT system for each alignment algorithm in use, we obtained the scores shown in Table 2. SPA substantially outperforms IBM4 when each is tuned – but not fully – to the translation input (as can be seen in the middle column), and performs marginally better (though not statistically significant) than IBM4 on unseen material for word-to-word alignments. When we see the EBMT performance with word and phrase level alignment and word-to-word level alignment, SPA is worse than SPA-W for the Devtest data set, but better for the Test data set. But for IBM4 and IBM4-W, we see IBM4 is better for the Devtest data set but worse for Test data set. Each difference was marginal and phrasal alignment didn't have big impact on EBMT performance and we need to investigate why phrasal alignment didn't improve EBMT performance in the future.

	Devtest	Test
EBMT	0.1563	0.13483
SPA	0.2259	0.17357
IBM4	0.2042	0.17085
SPA-W	0.2271	0.17324
IBM4-W	0.2019	0.17313

Table 2: French-English BLEU scores by algorithms

To see whether the performance is consistent, we made another test set ("test2") mostly drawn from file 040 and generated another set of reference sentences for the source sentences in the test2 data set such that each source sentence has two reference translations and got EBMT performance with each alignment algorithm. The results in Table3 show that IBM4 performs better than EBMT's internal aligner and SPA performs better than IBM4. The scores were expected to be higher than devtest with two reference translations.

	Test2
EBMT	0.2453
SPA	0.3027
IBM4	0.2666
SPA-W	0.3027
IBM4-W	0.2874

Table 3: French-English BLEU scores with two references

The performance of the EBMT-internal aligner is impacted to a large degree by lack of coverage in the lexicon it extracted from this rather

small training corpus – only two-third of the vocabulary has translations.

Our hypothesis was that improved alignment leads to improved translation. The experimental results clearly show that the Symmetric Alignment method does lead to better results than the EBMT system's own aligner, as well as the IBM model 4 alignments; however, the latter difference is not statistically significant on the evaluation set.

6. Future Work

The corpus used for these experiments was fairly small, in large part due to the computational expense of running the IBM Model 4 and SPA aligners – SPA can produce word-level and phrasal alignments for approximately 10,000 sentence pairs per hour. We intend to repeat the experiments with a larger training corpus.

To see if SPA substantially performs better than EBMT-internal aligner and IBM4, we are going to do more experiments on different language pairs and different domain data.

Further improvements may result from the generalized use of phrases and tighter integration with the dictionary generation process, as well as from updates to the EBMT code to permit exploitation of non-contiguous alignments.

7. References

Y. AL-ONAIZAN, J. CURIN, M. JAHR, K. KNIGHT, J. LAFFERTY, I.D. MELAMED, F.J. OCH, D. PURDY, N.A. SMITH, AND D. YAROWSKY. 1999. Statistical Machine Translation: Final Report. In Proceedings of the SummerWorkshop on Language Engineering. John Hopkins University Center for Language and Speech Processing.

RALF D. BROWN, REBECCA HUTCHINSON, PAUL N. BENNETT, JAIME G. CARBONELL, AND PETER JANSEN. 2003. Reducing Boundary Friction Using Translation-Fragment Overlap. In Proceedings of the Ninth Machine Translation Summit, pages 24-31, September. http://www.cs.cmu.edu/~ralf/papers.html.

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

R. FREDERKING, S. NIRENBURG, D. FARWELL, S. HELMREICH, E. HOVY, K. KNIGHT, S. BEALE, C. DOMASHNEV, D. ATTARDO, D. GRANNES, AND R. 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.

K. PAPINENI, S. ROUKOS, T. WARD, AND W. ZHU. 2002. BLEU: A method for automatic evaluation of machine translation. In ACL '02.