[*Translating and the Computer 23*: papers from the Aslib conference held on 29 & 30 November 2001]

Upping the Ante for "Best of Breed" Machine Translation Providers

Chris Callison-Burch
Amikai, Inc.
343 Vermont Street*
San Francisco, CA 94103, USA

October 7, 2001

Abstract

The notion of "best of breed" among value-added machine translation technology providers is generally defined as providing access to the single best commercially available machine translation engine for each language pair. This paper describes the efforts of Amikai, Inc. to go beyond that definition of best of breed. Rather than relying on a single engine for each pair, we have written a program that automatically selects the best translation from a set of candidate translations generated by multiple commercial machine translation engines. The program is implemented using a simple statistical language modelling technique, and relies on the simplifying assumption that the most fluent item in the set is the best translation. The program was able to produce the best translation in human ranked data up to 19% more often than the single best performing engine.

1 Introduction

Value-added machine translation technology providers do not create machine translation engines, but instead repackage and resell other companies' technology – adding something of value in the process. Amikai, Inc. is one such business. Amikai differentiates itself from machine translation engine vendors by:

- simplifying the process of integrating machine translation into existing code bases, and providing expert advice on how to do so

* The author is now at University of Edinburgh, Division of Informatics, 2 Buccleuch Place, Edinburgh EH8 9LW, UK. E-mail: ccb@csli.stanford.edu

- selling translation as a hosted service available over the internet
- scaling the translation engines so that they can handle millions of requests per day
- offering augmentations to the translation engines, such as customized dictionaries
- providing more complete coverage of language pairs, or higher quality engines for particular pairs, than available from a single vendor

There are a large number of machine translation engine vendors, some offering language pairs which others do not. Since machine translation resellers are not limited to engines from a single vendor, they are able to to assemble an extensive set of language pairs by combining engines from different vendors. Amikai provides access to all of its constituent engines through a uniform software interface, which abstracts away from the software calls associated with any particular engine. This abstract interface is advantageous not only because customers can translate between any of the available languages using a single interface, but also because it means that Amikai can essentially treat MT engines as modular components that can be easily replaced.

The modularity of MT engines has led to the use of the label "best of breed" among value-added machine translation technology providers. "Best of breed" is generally defined as providing access to the single best commercial machine translation engine on the market for each language pair. Since there are a number of translation engines for each language pair, Amikai does internal qualitative evaluations of the overall quality for each of the engines and licenses the engine which it deems the best for each pair.[1] For example, Amikai evaluated more than half a dozen Japanese ⇔ English translation engines before settling on the engine that it currently uses to provide translations between those languages.

The following sections detail Amikai's bid to go beyond the standard definition of best of breed. Rather than relying on a single engine for each pair, we have written a program that runs a number of translation engines in parallel and produces a set of possible translations. The program then automatically selects the best translation from the candidate set, and provides that translation to the user. Rather than evaluating each of the translation engines available for a particular language pair and relying on the one that is best *overall,* this program allows us to utilize the best possible engine *for every input.* Provided that the program successfully in selects the best translation from a set generated, then our translation quality is guaranteed to be at least as good as that of the best constituent engine. Provided that one

[1] Translation quality is something that is notoriously difficult to evaluate, and the matter of choosing the "best" translation engine is additionally complicated by the fact that quality is not the only dimension upon which a company like Amikai evaluates an engine. Salability, stability, time for translation, and licensing issues also come into play.

engine does not uniformly produce the best translation, then Amikai would in fact have better overall quality than any individual engine. That would be a clear value-add from our customers' perspectives, and would further distinguish Amikai's services.

2 Dynamically Choosing the Best Translation

The idea of building a program which dynamically selects the best translation from a set of translations begs the questions: Isn't being able to automatically distinguish the quality of translations essentially as difficult as building a machine translation engine? In that case, why use third party translation engines at all?

At first blush, it seems like selecting the best translation for a particular input from a set should be nearly as difficult as translating that input. You might imagine a strategy wherein the input is analyzed in some manner to construct a semantic representation or logical form of some sort, and each of the candidate translations are analyzed in a similar fashion. The semantic representations could then be compared using some sort of metric to find the translation that most closely matches the meaning of the input. That strategy is nearly as elaborate as building a machine translation engine (it only neglects the generation component). Building a fully-fledged translation engine is an enormous task. We've simplified the problem of choosing the best translation by making one crucial assumption: that the most fluent output corresponds to the best translation. With that assumption, the problem of choosing the best translation can be divorced from me input, and the problem is reduced to choosing the most well-formed example of the target language.

Assuming that the most fluent translation is the best translation isn't necessarily a well-grounded assumption — it would reward systems which had better generation components over those which actually preserve the meaning of the input. However, when translation engines fail to do a complete analysis of the input, their recovery strategies for producing a translation often result in "word salad". Therefore, if an engine produces a fluent translation then it is likely that the engine had a successful analysis of the input, increasing the likelihood that the meaning will be successfully transferred. We conducted an experiment, testing people's judgments on fluency and translation quality for machine translated sentences and found the two to be highly correlated as detailed in section 5.

2.1 Statistical Language Models

In order to automatically rank the translations that are produced by our collection of commercial translation engines, we assign a probability to each engine's output with a statistical language model of the target language. A language model judges the probability that each output is a sentence in the language that it models. The highest ranking output from the translation engines is deemed to be the most fluent, and therefore best translation. Statistical language models are not new to natural language processing. They are fundamental to speech and optical character recognition, and are used in spelling correction, handwriting recognition and statistical machine translation. For example, Hidden Markov Models have been applied with great success to speech recognition systems. Rather than relying on the raw speech signal to predict the next word, HMMs allow hypotheses to be generated about a series of words given the probabilities of the previous words. Word sequences are determined by finding the maximum probability path through the HMM (Ney 1998).

We built a language model for English using a web crawler to gather the text of 800 articles from the Internet magazine Salon. This corpus was augmented with 7,000 English inputs from Amikai chat rooms, and 12,000 English questions filtered from search data sent to a natural language search engine. The total size of the corpus was just over two million words. The statistical model that we generated was a simple trigram model with smoothing, following (Knight 1999). Other statistical models, such as a Hidden Markov Model, could have been used, but the trigram approach is simpler to implement and gives impressive results, as shown below.

To assign a probability to a sentence, a table was created recording the number of occurrences of every word, bigram (ordered word pair), and trigram (ordered word triple) in our corpus. These counts were used to assign a probability to each of those units in the sentence being evaluated. The probability of a word *x* occurring is the number of occurrences of *x* divided by the total number of words seen. The probability of a bigram *xy* is *p(y\x) = numOccurrences(xy)/numOccurrences(x).* The probability of a trigram *xyz* is *p(z|xy) = numOccurrences(xyz)/numOccurrences(xy).* Our program assigns a probability to each English sentence generated by a translation engine by taking the product of the probability of each of the sentence's trigrams.[2]

The result of using a trigram language model is that sentences with vocabulary and word ordering that are similar to the observed language are assigned a higher probability than sentences with strange word ordering or uncommon vocabulary. This corresponds fairly well to the intuitive meaning of fluency.

[2] The program actually smooths the trigram probabilities using the probabilities of the bigrams and words and a very small constant to counteract the effects of sparseness in the data, which would frequently cause a probability of zero for unobserved word sequences. See (Callison-Burch and Flournoy 2001) for details.

2.2 A Concrete Example

To illustrate this, let's say the Japanese input "ashita kara haruyasumi" is fed into the program. The program would then produce a candidate set of translations from four separate Japanese to English translation engines. The set of translations produced for this sentence is:

$$\left\{\begin{array}{l}\text{"Spring it is a day off from tomorrow.",}\\ \text{"It is the spring vacation from tomorrow.",}\\ \text{"It will be haruyasumi from tomorrow.",}\\ \text{"It is spring vacation from tomorrow."}\end{array}\right\}$$

The probability of each of the sentences would then be calculated based on the produce of its trigrams:

P(Spring it is a day off from tomorrow) =
p(Spring start-of-sentence start-of-sentence)*
p(it | start-of-sentence Spring)*
p(is | Spring it)*
p(a | it is)*
p(day | is a)*
p(off | a day)*
p(from | day off)*
p(tomorrow | off from)*
p(end-of-sentence | from tomorrow)*
p(end-of-sentence | tomorrow end-of-sentence)

The program then consults its frequency tables (which were precompiled from the corpus of English text) to calculate the value of each multiplicand, and assigns a probability to the sentence. It repeats this process for each of the other three sentences, and then ranks the sentences in order according to their probabilities. In this case the program ranked them in this order:

1. "It is spring vacation from tomorrow."
2. "It is the spring vacation from tomorrow."
3. "Spring it is a day off from tomorrow."
4. "It will be haruyasumi from tomorrow."

And thus would return "It is spring vacation from tomorrow." as the translation of the Japanese input, discarding the other translations.

3 Evaluation

The performance of the program was evaluated against a set of hand-ranked data. Sets of machine-translated English sentences were produced from Japanese and French sentences using multiple translation engines. For each Japanese sentence, a set of four English sentences was produced. For each French sentence, a set of two English sentences was produced. Each of the translated sentences was assigned a fluency rating by a monolingual English speaker using the following scale:

1. Nearly Perfect - the sentence is a fluent English sentence.
2. Understandable - the sentence is understandable but may have (slightly) strange word choice, or contain some minor grammatical errors, such as an incorrect preposition or determiner.
3. Barely Understandable - the sentence contains several grammar and/or vocabulary errors and can only be understood with great effort on the part of the reader.
4. Incomprehensible - the meaning of the sentence cannot be derived.

The program was run on each of the sets of translations. The program assigned probabilities to each sentence in a set, and returned the sentence that it found to have the highest probability, as described in section 2.2. The program's performance is given in the following tables as a percentage that is calculated by dividing the number of sets that were considered by the number of times that the program successfully picked out the highest human ranked sentence in a set (or, in the case of ties, one of the sentences which received the highest rank),

In comparing the model performance to the human rankings, we considered the baseline measure to be the single engine which received the most top ranks from the human subjects. If our program did not perform better than this baseline then there would be no point in integrating it into our translation architecture - the baseline essentially measures the current best of breed translation technology.

4 Results

The results are separated based on the types of inputs that the candidate sets were generated from. For the translations into English, 100 Japanese sentences were gathered from Amikai's chat rooms, 154 French sentences were taken from web pages, and 84 French sentences were taken from the chat rooms. Table 1 shows the performance of our program and each of the four translation engines that were used to generate the candidate sets of translations for the Japanese chat sentences. In this case the baseline engine was Engine B,[3] which produced the highest ranked candidate translation 70% of the time overall. Our program outperformed the baseline engine, choosing the highest ranked translation 4% more often.

We note that the performance of our program is most important when the candidate translations are understandable or nearly perfect, because distinguishing between the better of barely understandable sentences does not

[3] The translation engine names have been removed because Amikai treats evaluation information as proprietary.

	All Inputs (100 sets)	At Least *Understandable* (86 sets)	*Nearly Perfect* (57 sets)
Multi-engine comparison	*74%*	*73%*	*77%*
Engine A	58%	54%	61%
Engine B	70%	*69%*	*66%*
Engine C	27%	21%	19%
Engine D	40%	35%	39%

Table 1: Japanese to English Chat Sentences

	All Inputs (154 sets)	At Least *Understandable* (118)	*Nearly Perfect* (38 sets)
Multi-engine comparison	*84%*	*81%*	*87%*
Engine E	*76%*	*70%*	*68%*
Engine F	58%	52%.	45%

Table 2: French to English Web Page Sentences

increase the usability of our products as much. Therefore, we refined the performance results by grouping the sets according to their highest ranked translation. Performance values are given for all sets, for those sets which contained a sentence rated *Understandable* or higher, and for those sets which contained a sentence rated *Nearly Perfect.* In this case, the relative performance of our program was higher than the baseline engine on those sets than it was for all sets. For the 57 sets which contained at least one sentence which had been ranked *Nearly Perfect*, the baseline engine got the rank 66% of the time, whereas our program was able to select the best sentence 77% of the time – a 11% increase over the baseline as opposed to the 4% overall.

Table 2 shows the performance for two translation engines which translated sentences gathered from French web pages, and for our program selecting between the candidate translations generated by those two engines. Our multi-engine comparison program again outperformed the baseline engine, scoring 8% better overall, 11% better on the 118 sets which contained understandable translations, and 19% higher for the 38 sets which contained a nearly perfect translation.

Interestingly, the baseline engine changed from Engine E to Engine F for translations of French chat room sentences, as shown in Table 3. This is evidence that some MT systems perform better for certain types of language usage than other systems do, and perform worse than those systems do for

	All Inputs (84 sets)	At Least *Under-standable* (61 sets)	*Nearly Perfect* (34 sets)
Multi-engine comparison	*94%*	*92%*	*100%*
Engine E	71%	66%	68%
Engine F	*86%*	*80%*	*85%*

Table *3:* French to English Chat Sentences

other types of language usage, which was a motivating factor for creating our program. For the relatively informal constructions found in chat, Engine F received the highest rating 86% of the time overall, and 85% of the time (compare to Engine E's 68%) for sets which contained at least one nearly perfect translation. The multi-engine comparison program yielded high results (probably due in part to the large number of ties between the fluency ratings of the two candidate translations in each set). The program choose the best translation more than 90% of the time for all sets, and 100% of the time for the 34 sets containing a nearly perfect translation.

4.1 A Non-English Model

In order to test that the statistical language modelling technique employed by our program would also be effective for languages other than English, we built a language model for French using a corpus containing 1.1 million words. Using English sentences gathered from web pages and chat rooms, we created candidate translation sets using five translation engines (Engines E, F, and three others labeled G-I). The machine generated French sentences were then ranked by a French speaker using the same scale as for the English sentences. Table 4 shows the results for the program using the French language model.

The multi-engine comparison tool again performed better than the baseline engine - though to a lesser extent than in previous tests - but still enough to suggest that the technique has the potential to transfer to other languages. Because the French language model was built from a corpus only about half the size of the English corpus, and because French has a richer morphology than English (leading to a higher number of word forms), there is a decreased likelihood than a word in a candidate translation would have been observed in the training corpus. The fact that the program did outperform the baseline even given these constraints bodes well for the method. Its performance may well increase to the same levels as for the English, given an increased corpus size.

	All Inputs (51 sets)	At Least *Understandable* (44 sets)	*Nearly Perfect* (34 sets)
Multi-engine comparison	67%	*61%*	*64%*
Engine E	53%	48%	47%
Engine F	49%	41%	47%
Engine G	45%	36%	32%
Engine H	51%	45%	44%
Engine I	*63%*	*57%*	62%

Table 4: English to French Sentences (Chat and Web Page)

5 Verification of the Assumption

In order to test the assumption that the most fluent output of the machine translation engines corresponds to the best translation, we designed an experiment to compare how people rate fluency to how they rank translation quality.

For the experiment we had a group of nine bilingual subjects rate a subset of the data that we used to determine the program's performance. The experiment was divided into two parts. For the first part the translations were displayed in random order without showing the source text. The subjects were asked to rate the fluency of each of the sentences according to the previously mentioned scale. For the second part, the subjects, who were fluent in either French or Japanese, were shown sets of translations paired with the original sentence. They were asked to rank the sentences in each set based on the quality of the translation.

A within-subject comparison was done between each subject's fluency rating and his or her translation quality rankings for each sentence. The relative ordering for each pair of translations in a set was compared. If the subject assigned the same relative ranking to a pair of translations for both the fluency and the translation quality tests, then we counted that as a match. For cases where one of the tests was judged a tie, we used either a strict or loose method for comparison. In the loose method for comparison, if the ratings were tied for one test but not the other, we counted a match. In the strict method, we counted a tie as a match only if the scores were also tied in the other test. The similarity was calculated by dividing the number of matches by the total number of comparisons.

Using the strict comparison method the subjects had an average of 90.7% similarity between their fluency and translation quality scores. Using the loose method for comparison that number increased to 99.39%. We took this to be strong evidence that our simplifying assumption was well founded.

[References absent from original paper]