A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines

Chris Callison-Burch, Raymond S. Flournoy

Amikai, Inc.
343 Vermont Street
San Francisco, CA 94103
USA
ccb@amikai.com, rflournoy@amikai.com

Abstract

This paper describes a program that automatically selects the best translation from a set of translations produced by multiple commercial machine translation engines. The program is simplified by assuming that the most fluent item in the set is the best translation. Fluency is determined using a trigram language model. Results are provided illustrating how well the program performs for human ranked data as compared to each of its constituent engines.

Keywords

machine translation, language modeling, statistical methods, translation quality

Introduction

Amikai, Inc. develops machine translation applications. The technology is a combination of in-house and out-sourced material, with the bulk of the translation work coming from various third party translation engines. Amikai is a "best of breed" provider, which means that for each language pair that we support we provide access to the (single) best translation engine available on the market.

This paper describes our efforts towards developing a system which uses multiple translation engines for each language pair, and dynamically chooses the best translation from a candidate set of translations for each input. We reasoned that if we built a program that could identify the best translation within a set then we would be able to claim that our quality was at least as good as the best translation engine used to produce the candidate set. Provided that one engine did not uniformly produce the best translation, then we would in fact have better overall quality than any given engine.

The idea of building such a program 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? 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 the input, and the problem is reduced to choosing the most fluent output. When translation engines fail to do a complete analysis of the input, their recovery strategies for producing a translation often result in "word salad". 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 verified our assumption with human testing, and built a statistical language model to rank fluency automatically.

Prior Work

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. Each language model judges the probability that each output is a sentence in that language. The highest ranking output 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[1]. 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).

In the early 1990s, Brown et al. applied statistical modelling to machine translation. By inducing word alignments and "fertilities" from parallel bilingual corpora, Brown et al. were able to produce a translation model for the aligned languages. This model allows a probability P(e | f) to be assigned to a pair of sentences, the French sentence *f* and the English sentence *e*, that *e* is a translation of *f*. The best translation is the sentence which maximizes P(e | f) or P(e) * P(f | e) using Bayes' Rule. P(e) represents the probability that *e* is an English sentence, and is used to generate translations which are natural and grammatical. A language model built from a monolingual corpus can be used to assign this probability, and P(f | e) is determined using the bilingual corpora alignments.

The English sentence which maximizes both P(e) and P(f | e) will be the best translation of *f* . The design of our program essentially assumes that all the machine translation engines produce similar P(f | e) values, and looks for the value which has the highest P(e) score.

[1] For an overview of these see Manning and Schütze (1999).

A Trigram Language Model

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 still 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 a sentence. 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 b(y | x) = number-of-occurrences("xy") / number-of-occurrences("x"). The probability of a trigram xyz is b(z | x y) = number-of-occurrences("xyz") / number-of-occurrences("xy"). The probability of a sentence could be calculated based on its trigrams as follows:

P(I like snakes that are not poisonous) ~
b(I | start-of-sentence start-of-sentence) *
b(like | start-of-sentence I) *
b(snakes | I like) *
...
b(poisonous | are not) *
b(end-of-sentence | not poisonous) *
b(end-of-sentence | poisonous end-of-sentence)

Our program assigns a probability to each English sentence by taking the product of the probability of each of its trigrams, but smoothes those trigrams with the probabilities of the bigrams and words to counteract the effects of sparseness in the data.

Instead of

b(z | x y) = number-of-occurrences("xyz")/
number-of-occurrences("xy")

it uses

b(z | x y) = 0.80 *number-of-occurrences("xyz")/
number-of-occurrences("xy")+

0.14 * number-of-occurrences ("yz")/
number-of-occurrences ("z")+
0.099 * number-of-occurrences("z")/
total-words-seen + 0.001

These coefficients were determined by training on a subset of the human ranked data[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.

Results

The performance of the program was rated using data collected from three types of sources: Japanese chat rooms, French chat rooms, and French web pages. Each of the sentences from these sources was translated into a set of English sentences using commercial machine translation engines. For translation from Japanese into English, four engines were used. They are labeled Engines A-D[3]. For translations from French into English two engines were used (Engines E, F).

The fluency of each translation was ranked by a monolingual English speaker according to the following scale:

1 : *Almost Perfect* – the sentence is a fluent English sentence. It seems like it was written by a native speaker.
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 then run on each set of translations, and returned the sentence that it rated as the most fluent. The program was awarded a point each time the sentence that it selected was also the most highly ranked for that set by the human subject. The program's performance is given below as a percentage, which is calculated by dividing its points by the number of sets. The engines' scores are determined similarly. Because of ties, the engine scores do not sum to 100%.

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 what is considered to be "best of breed" translation technology.

We note that the performance of the program is most important when the candidate translations are understandable or nearly perfect, because distinguishing between the better of barely understandable sentences does not 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 at least one sentence which was rated *Barely Understandable* or higher, for those sets which contained a sentence rated at least *Understandable*, and

[2] The details of the coefficient determination are omitted for brevity.

[3] The specific engine names have been removed.

for those sets which contained a sentence rated *Almost Perfect*.
Here are the results that we obtained:

	All (100 sets)	*At least Barely Understandable* (94)	*Understandable* (86)	*Almost Perfect* (57)
Multi-engine Tool	**74%**	**72%**	**73%**	**77%**
Engine A	58%	55%	54%	61%
Engine B (baseline)	**70%**	**68%**	**69%**	**66%**
Engine C	27%	22%	21%	19%
Engine D	40%	36%	35%	39%

Table 1: Japanese→ English chat translations

In Table 1, the baseline engine produced the highest ranking candidate translation 70% of the time overall, 68% of the time for translations which were at least barely understandable, 69% for translations which were at least understandable, and 66% for translations which were almost perfect. Our multi-engine comparison tool outperformed the baseline engine, choosing the best translation 4% more often for all sets, and 11% more often for translations which were nearly perfect.

	All (154 sets)	*At least Barely Understandable* (146)	*Understandable* (118)	*Almost Perfect* (38)
Multi-engine Tool	**84%**	**82%**	**81%**	**87%**
Engine E (baseline)	**76%**	**75%**	**70%**	**68%**
Engine F	58%	56%	52%	45%

Table 2: French → English web page translations

For the French to English translations of sentences from web pages, the baseline engine was Engine E. Again the multi-engine comparison tool outperformed the baseline, with scores ranging from 7% to 19% higher than the baseline at choosing the best translation.

	All (84 sets)	*At least Barely Understandable* (61)	*Understandable* (61)	*Almost Perfect* (34)
Multi-engine Tool	**94%**	**92%**	**92%**	**100%**
Engine E	71%	66%	66%	68%
Engine F (baseline)	**86%**	**80%**	**80%**	**85%**

Table 3: French → English chat translations

For the French to English chat translations, the baseline engine was Engine F (Engine E performed worse for this type of informal language usage) which had the translations that were judged best around 86% of the time. The engine comparison tool had very high performance on this data set. It was able to pick the best translation more than 90% of the time, and was able to pick the best translation 100% of the time for the 34 candidate sets which contained a translation judged to be almost perfect.
We performed another test using translations from English into French using a trigram language model of French built from a corpus of a little over 1.1 million words. The English sentences were also gathered from web pages and chat rooms, and translated by five translation engines (Engines E, F, and G-I).

	All (51 sets)	*At least Barely Understandable* (50)	*Understandable* (44)	*Almost Perfect* (34)
Multi-engine Tool	**67%**	**66%**	**61%**	**64%**
Engine E	53%	52%	48%	47%
Engine F	49%	48%	41%	47%
Engine G	45%	44%	36%	32%
Engine H	51%	50%	45%	44%
Engine I	**63%**	**62%**	**57%**	**62%**

Table 4: English → French translations

The multi-engine comparison tool again performed better than the baseline engine, suggesting that this technique transfers well to other languages.

Human rankings

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 9 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 previous 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 rating for each sentence. The relative ordering for each pair of translations in a set was compared. If the subject assigned the same relative ranking for 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 as well. 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.

Conclusion

In this paper we described Amikai's system for dynamically choosing the best translation from a collection of commercial machine translation engines. Relying on the verified assumption that the best translation was generally the most fluent output from the engines, we were able to construct our program using a simple statistical language modeling technique. Furthermore, that technique is independent of the language being tested, and can easily be applied to other languages, or optimized to particular types of language usage. Our program performed up to 19% better than the baseline metric which was chosen to reflect the notion of best of breed for value-added machine translation technology providers.

References

Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer (1993). The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2).

Knight, K. (1999). A Statistical MT Tutorial Workbook manuscript prepared in connection with the Johns Hopkins University summer workshop.

Manning, C., and H. Schütze (1999). Foundations of Statistical Natural Language Processing: pp. 191-220.

Ney, H., ed. (1998). "Language Models." in Spoken Language Characterization:. pp. 91-140.

Salon. www.salon.com.