CS 224N

Programming Assignment 1

Gil Shotan - gilsho@stanford.edu

Rafael Ferrer - rmferrer@stanford.edu

Part 1: Alignment Models

Convergence Criterion

For simplicity of implementation we decided to use a fixed number of iterations to terminate the EM algorithm. The problem was to decide the “best” number of iterations.

We tried out different iteration limits for the EM algorithm and chose the one that gave the best AER scores. Additionally, we also experimented with the training set size as another optimization parameter.

The following table shows our AER error rates for IBM Models 1 and 2 on the (French, English) corpus for different training set sizes and different EM iteration numbers.

Model #	Max Iter.	# of Sentences	AER
1	10	1000	0.4507
1	10	10000	0.3544
2	10	1000	0.4406
2	10	10000	0.3140
1	20	1000	0.4527
1	20	10000	0.3544
2	20	1000	0.4359
2	20	10000	0.3133

Table 1: Effect of Iterations and Training Set Size

As can be seen from the results, it seems the choice of maximum number of iterations does not have a large effect at these levels, so we chose 10.
The size of the training set seems to be very important though, so all subsequent tests were run using the full training set size of about 10,000 sentences, unless otherwise noted.
This makes intuitive sense, as the probability estimates learned from data are going to improve with more of it.

Generalizing Across Languages

Development Set

Using 10,000 training sentences (Though the Hindi training set only has 3441 sentences)

AER	French-English	Hindi-English	Chinese-English
PMI	0.7127	0.8594	0.8251
Model 1	0.3544	0.5823	0.5870
Model 2	0.3133	0.5781	0.5794

Table 2: AER for Different Language Pairs - Development Set

Test Set

Again, using 10,000 sentences for each.

AER	French-English	Hindi-English	Chinese-English
PMI	0.6914	0.8229	0.8087
Model 1	0.3496	0.5802	0.5863
Model 2	0.3178	0.5815	0.5670

Table 3: AER for Different Language Pairs - Test Set

As can be clearly seen, the models work fairly well for French, a language that shares many key features with English. Unfortunately as one moves into languages with different grammar rules, some of the inherent assumptions break down. Some theories on why this would happen are presented here.

Hindi

Language word order is likely to play a big role in the confusion.
English, along with 42% of the worlds' known languages presents itself in the subject-verb-object order in sentences. "Sam ate oranges" sounds natural.
Hindi, however, is a subject-object-verb type language (that accounts for 45% of the world's languages). Hence the equivalent of "Sam oranges ate" is natural-sounding.

As one can imagine, this causes problem in an alignment model that assumes that words in the source and target languages should be aligned at similar positions. That being the essence of the “lying along the diagonal” concept in IBM Model 2.

Chinese

Here, word order plays a role too as Chinese is flexible enough to be found in any of the three word order types one considers.
More than that though, since characters in Chinese correspond to words, it’s possible that phrase-based algorithms have problems operating in Chinese.

Error Analysis and Improvement Ideas

IBM Model 1

A fairly frequent issue that would crop up while looking at alignment maps generated by an IBM Model 1 run is that many target language words would be aligned to the same source language word. Represented by a long horizontal line on the maps, this issue probably arises due to translation probabilities - $t(e_i|f_j)$ - that are artificially boosted since the source language word itself is pretty rare.
Accounting for vast vocabularies in NLP problems is an ongoing issue, and there will be variations in the frequency of words (Zipf’s Law).

IBM Model 2

On the whole, IBM Model 2 did much better than IBM Model 1.
Specifically we checked that the `<null>` token was working as expected, and it was. Target words were generated successfully in many cases.
A particular shortcoming was the focus on absolute position as a heuristic to learn the alignment probabilities. As explained previously, word order differs between languages and thus, there is no guarantee that an alignment should “lie along the diagonal”.

Some ideas for improvement include:

- Support many-to-many alignments rather than one-to-many. One-to-many does work when a source word maps to multiple target words, but the vice versa is not possible. For example, let English be our source language and Turkey be our target language. If inside of source sentence “I went” maps to “Gittim” in the target language, model 2 would not be able to support it. Many-to-many mappings would thus increase the number of correct alignments. This solution could be implemented with Model 3 (see discussion below about Model 3). Model 3 is better than model 2 because it handles many-to-many alignments.
- In the HMM Model, as an alignment heuristic they actually record probabilities for the difference in position between source and target words. This makes a good deal more

sense where you assume that phrase pairs are more likely to group together in a similar pattern in a language-independent fashion.

- To generalize better to other languages, we could also try taking into account the word order differences during alignment with an explicit parameter that specifies the languages being used. Then with the help of an accurate part-of-speech tagger we could use the information to learn better alignments.

Part 2: Machine Translation

In your report, please include the BLEU scores of both the baseline and the system with your feature. Identify a few interesting differences in the translation output. Give a few sentences of motivation for your feature. What property of the data were you targeting?

In this second part of the assignment, we implemented a numeric feature for a

Feature Set	**BLEU Score**
Baseline	15.099
Baseline + Numeric	15.376

Weight: -12.147