Kat Busch
Veni Johanna
CS224n Programming Assignment 1

Division of labor: Kat worked mostly on PMI and Part 2, Veni on model 1 and 2

Part 1: Alignment

PMI Model

Development Set

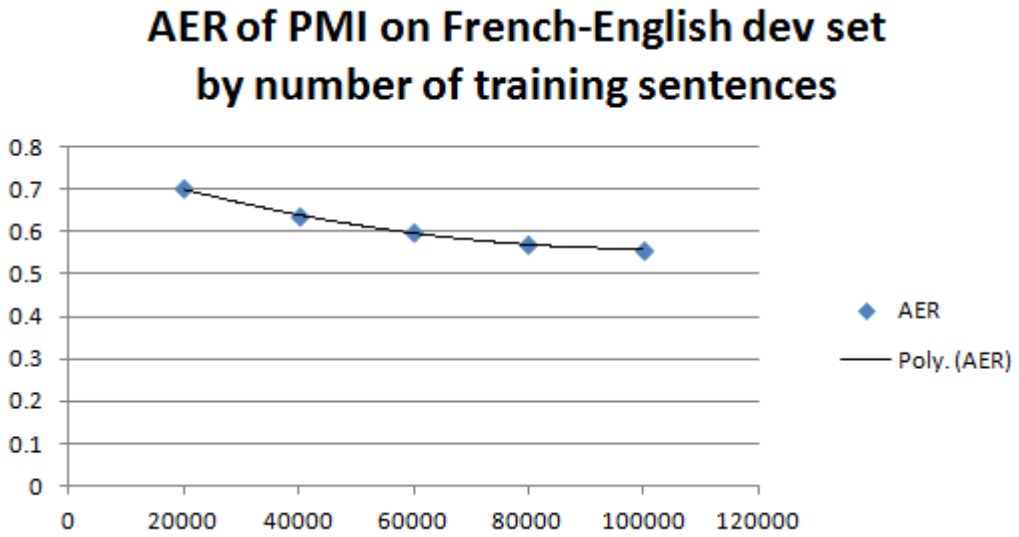
# of Training Sentences	French-English	Hindi-English	Chinese-English
10,000	Precision: 0.2386 Recall: 0.2160 AER: 0.7686	Precision: 0.1487 Recall: 0.1286 AER: 0.8621	Precision: 0.1617 Recall: 0.1137 AER: 0.8665
20,000	Precision: 0.2982 Recall: 0.3018 AER: 0.7007	N/A	Precision: 0.1811 Recall: 0.1274 AER: 0.8504
30,000	Precision: 0.3440 Recall: 0.3609 AER: 0.6506	N/A	Precision: 0.1946 Recall: 0.1369 AER: 0.8393
40,000	Precision: 0.3564 Recall: 0.3817 AER: 0.6355	N/A	Precision: 0.2029 Recall: 0.1427 AER: 0.8324
50,000	Precision: 0.3814 Recall: 0.4024 AER: 0.6119	N/A	Precision: 0.2133 Recall: 0.1500 AER: 0.8239

Test Set:

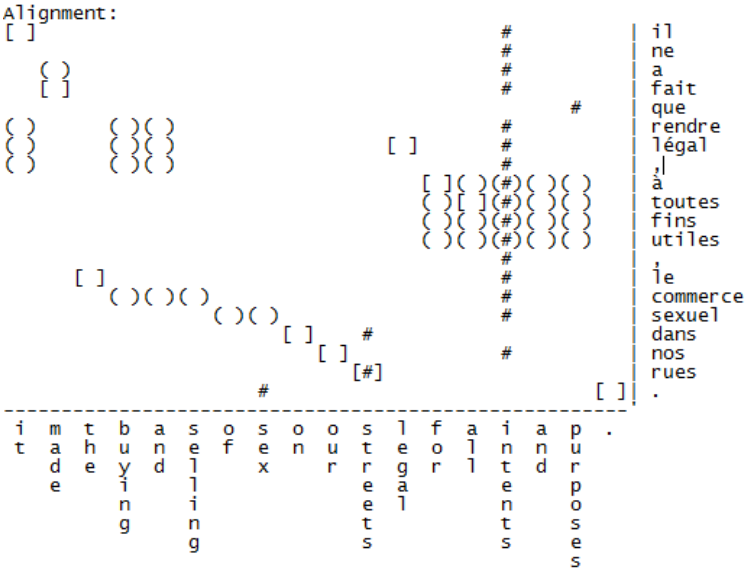
# of Training Sentences	French-English	Hindi-English	Chinese-English

10,000	Precision: 0.2996 Recall: 0.2952 AER: 0.7019	Precision: 0.1633 Recall: 0.1534 AER: 0.8418	Precision: 0.1908 Recall: 0.1357 AER: 0.8414
20,000	Precision: 0.3434 Recall: 0.3606 AER: 0.6507	N/A	Precision: 0.2101 Recall: 0.1494 AER: 0.8254
30,000	Precision: 0.3742 Recall: 0.4049 AER: 0.6153	N/A	Precision: 0.2242 Recall: 0.1594 AER: 0.8136
40,000	Precision: 0.3940 Recall: 0.4331 AER: 0.5926	N/A	Precision: 0.2350 Recall: 0.1671 AER: 0.8047
50,000	Precision: 0.4070 Recall: 0.4507 AER: 0.5780	N/A	Precision: 0.2444 Recall: 0.1738 AER: 0.7969

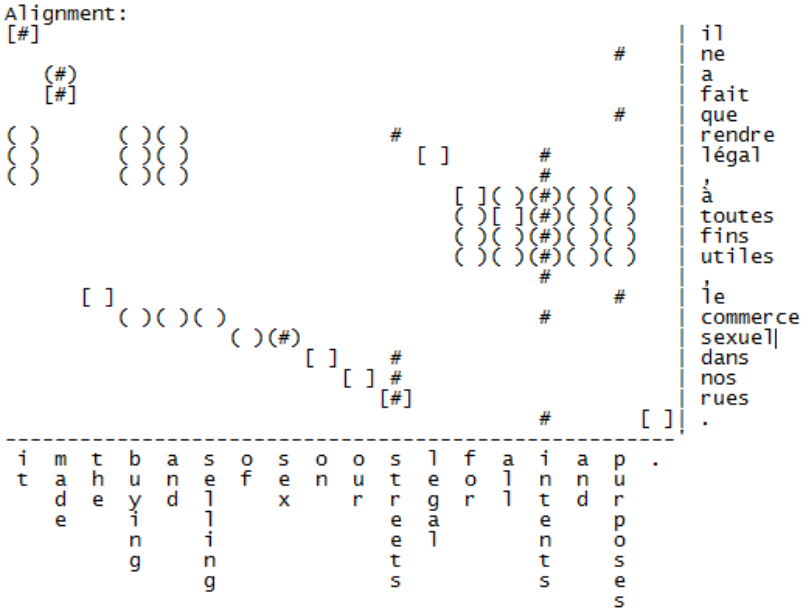
PMI results increase steadily with the amount of training data but seems to level off.



The model suffers a lot from sparsity and is particularly sensitive to rare words. For instance, in the following clearly failing alignment, since intents appears so infrequently in the data, its multiplier in the denominator P(intents) is very low, so its probability appears the highest match for almost every target word.



However, with more samples of each word it can more easily associate the word with others that co-occur with it. Here is the same alignment with 100,000 training examples:



The model is able to make slight improvements with a bunch more training data, but it is still not very good. It is still very sensitive to the frequency of the words over the co-occurrence and has no concept of the “diagonal line” that alignments should take.

Model 1

Development Set:

# of Training Sentences	French-English	Hindi-English	Chinese-English
10,000	Precision: 0.5312 Recall: 0.6923 AER: 0.4174	Precision: 0.4628 Recall: 0.4004 AER: 0.5706	Precision: 0.4605 Recall: 0.3239 AER: 0.6197
20,000	Precision: 0.5548 Recall: 0.7219 AER: 0.3919	N/A	Precision: 0.4713 Recall: 0.3315 AER: 0.6108
30,000	Precision: 0.5645 Recall: 0.7278 AER: 0.3834	N/A	Precision: 0.4788 Recall: 0.3367 AER: 0.6046
40,000	Precision: 0.5659 Recall: 0.7337 AER: 0.3805	N/A	Precision: 0.4830 Recall: 0.3398 AER: 0.6011
50,000	Precision: 0.5756 Recall: 0.7396 AER: 0.3720	N/A	

Test Set:

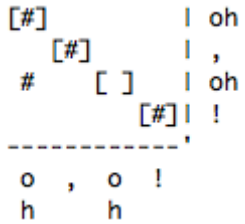
# of Training Sentences	French-English	Hindi-English	Chinese-English
10,000	Precision: 0.5573 Recall: 0.7358 AER: 0.3816	Precision: 0.4475 Recall: 0.4205 AER: 0.5665	Precision: 0.4530 Recall: 0.3221 AER: 0.6235
20,000	Precision: 0.5734 Recall: 0.7538 AER: 0.3649	N/A	Precision: 0.4645 Recall: 0.3303 AER: 0.6139
30,000	Precision: 0.5857 Recall: 0.7662 AER: 0.3525	N/A	Precision: 0.4686 Recall: 0.3332 AER: 0.6105
40,000	Precision: 0.5934 Recall: 0.7729 AER: 0.3452	N/A	Precision: 0.4749 Recall: 0.3376 AER: 0.6053
50,000	Precision: 0.5980 Recall: 0.7764 AER: 0.3410	N/A	

The precision and recall of Model1 results generally increase (hence the error rate decreases) as the number of training sentences increases. They are also significantly and consistently better than the PMI result.

Our Model1 uses a simple heuristic to define convergence - in the end of each training iteration, we compare the new value of *t* to the old value of *t*. We consider a probability *converged* if the absolute difference of the two values is less than 10^{-6}, and the parameter *converged* if more than 95% of the *t* probabilities are converged. Hence, the number of iteration depends on each development set and not on a rigid criterion.

As a memory optimization, our parameter initialization of Model1 doesn't initialize *t(e | f)* for all e and f in the training set vocabulary. Instead, we initialize *t (e | f)* for all e that occur together with f the development set. Normalization is done in the end of each training iteration. To make sure that this normalization is done right, after the normalization step is done, we check that conditional probabilities add up to 1 at the end of every other training iteration.

Many of the mistakes in Model1 come from duplicate words in a sentence. A really simple example:



Here, the model correctly matches the word "oh" in French to "oh" in English, however it incorrectly aligns the second "oh" to the first, instead of the second, "oh". More similar mistakes come from multiple punctuations in a sentence.

Model 2

Development Set:

# of Training Sentences	French-English	Hindi-English	Chinese-English
10,000	Precision: 0.5853 Recall: 0.7160 AER: 0.3730	Precision: 0.4580 Recall: 0.3963 AER: 0.5751	Precision: 0.5245 Recall: 0.3689 AER: 0.5668
20,000	Precision: 0.6117 Recall: 0.7604 AER: 0.3409	N/A	Precision: 0.5331 Recall: 0.3750 AER: 0.5597
30,000	Precision: 0.6241 Recall: 0.7692 AER: 0.3296	N/A	Precision: 0.5396 Recall: 0.3795 AER: 0.5544
40,000	Precision: 0.6186 Recall: 0.7663 AER: 0.3343	N/A	Precision: 0.5438 Recall: 0.3825 AER: 0.5509

50,000		N/A	

Test Set:

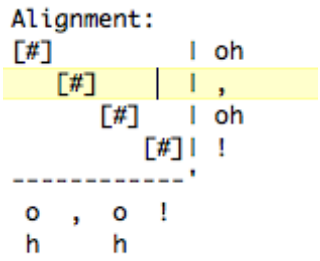
# of Training Sentences	French-English	Hindi-English	Chinese-English
10,000	Precision: 0.6396 Recall: 0.7878 AER: 0.3097	Precision: 0.4271 Recall: 0.4013 AER: 0.5862	Precision: 0.5044 Recall: 0.3586 AER: 0.5808
20,000	Precision: 0.6579 Recall: 0.8108 AER: 0.2898	N/A	Precision: 0.5139 Recall: 0.3654 AER: 0.5729
30,000	Precision: 0.6694 Recall: 0.8185 AER: 0.2796	N/A	Precision: 0.5196 Recall: 0.3695 AER: 0.5681
40,000	Precision: 0.6780 Recall: 0.8227 AER: 0.2725	N/A	Precision: 0.5249 Recall: 0.3732 AER: 0.5638
50,000	Precision: 0.6823 Recall: 0.8262 AER: 0.2685	N/A	

The implementation of Model2 is quite similar to Model1, with the exception of an additional parameter, the alignment probability *q.* This parameter is initialized uniformly and recomputed in the end of each training iteration. Similarly to Model1, as a memory optimization, our parameter initialization of Model2 doesn't initialize *t(e | f)* for all e and f in the training set vocabulary. Instead, we initialize *t (e | f)* for all e that occur together with f the development set. This parameter is initialized with the result of Model1 training.

Our implementation of Model2 also uses a similar heuristic to Model1 to define convergence - in the end of each training iteration, we compare the new value of *t* to the old value of *t*. We consider a probability *converged* if the absolute difference of the two values is less than 10^{-6}, and the parameter *converged* if more than 95% of the *t* probabilities are converged. As an addition, we capped the number of iteration to 50 to avoid the algorithm to run too long. Normalization is done in the end of each training iteration. To make sure that this normalization is done right, after the normalization step is done, we check that conditional probabilities add up to 1 at the end of every other training iteration.

The precision and recall of Model2 results generally increase (hence the error rate decreases) as the number of training sentences increases. They are also consistently better than the Model1 result. However, the AER for 40,000 French sentences is greater than 30,000 French sentences - this anomaly might be caused either by the convergence cap.

Since Model2 keeps track of alignment probability, it mostly solves the problem with Model1 outlined above. The exact same test case, for example, gives the following correct result:



General Observations

The result across languages differs pretty significantly - French-English result is consistently better than both Hindi-English and Chinese-English in all models. This is predictable, since the two languages are somewhat closely related; “French is a Latin language with German and English influence, while English is a Germanic language with Latin and French influence” (french.about.com). On the other hand, Hindi / Chinese and English belong to different language families.

In the Chinese data set, many of the mistakes came from verb inflections, such as is/was/are or make/makes/made, since the information contained by verb inflections in English is conveyed mostly through word order in Chinese. We suspect that the result improvement in Model2 from Model1 is much more visible in Chinese-English than in French-English because of this language difference. Since Model 2 takes into account the alignment probability in addition to probability that’s purely based on cooccurrence count of words in Model1, Model2 can take into account the prominence of word order in Chinese better than Model1.

Most of the mistakes in the French-English alignment seem to lie at prepositions and articles (*a, the, le, de*). Errors in punctuations seem to be a low-hanging fruit that small modifications can improve.

Part 2: MT System Tuning

For system tuning, we used Model 2 (initialized with Model 1) on a set of 50,000 alignment training examples. We achieved a highest Bleu score of 15.24. For comparison and sanity checking, we also tested several different with several phrase tables generated from only 20000 alignments, and these (after tuned with MERT) resulted in Bleu scores in the mid-13 range. It is expected that only 20000 would perform decently worse because of much increased sparsity, and the test confirmed those findings. The rest of the Moses runs discussed here use the 50000 input alignments.

Section 1: Phrase Table

The phrase table itself reveals a lot about how the translations take place. Though we have a sample size of only 2 for each run, it appears that runtime increases with maximum phrase

length as would be expected. This is helped by the very small deviations between the two runs that we timed. They were delta of 0 seconds, 2 seconds, and 7 seconds for 4, 6, and 8 respectively. As there are longer phrases, there are more phrases to consider that were previously cut off for being too long.

Max Phrase Length	4	6	8
Runtime (s)	88	154	202.5

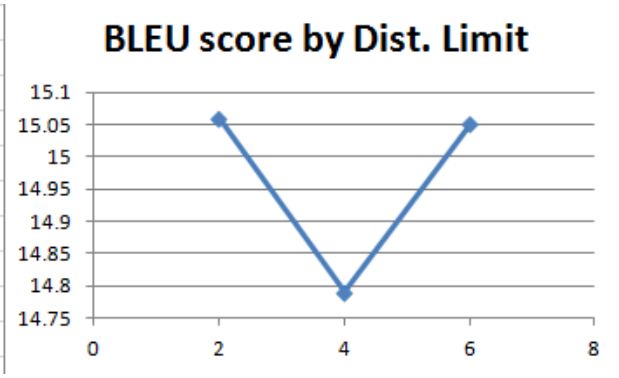
I have chosen to focus on the phrase table with a maximum phrase of 6 because that was the default in the assignment and our mid-scoring length. The phrase table has 1474413 phrasal alignments. The first thing to note is that the phrase table is, of course, huge. There were only 50000 sentences going into to it, so it considers many possible phrases. Of those it considers, some seem very accurate. For instance, the phrase “la chine avait acquis” maps with “China has acquired.” “De la part du” goes to both “on the part of” and “on the part of the” which is likely from how it chunks the phrases. There are some relics of worse chunkings. “Mais” translates appropriately to “but,” yet it also goes to “conflict, but” and “but tolerating.” The apostrophe translates to many, many things from missile to new deal. Some sort of understanding of what punctuation tokens are would likely benefit the system.

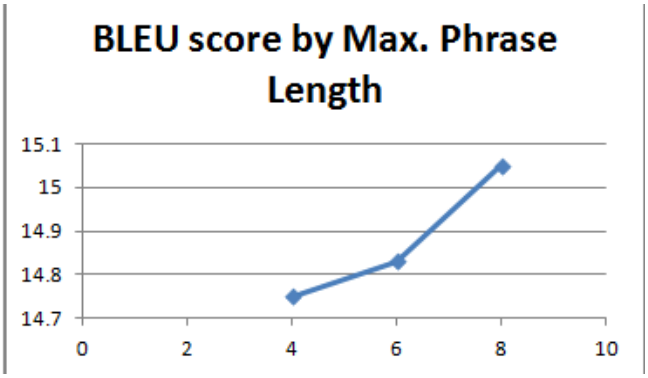
Section 2: Tuning

Phrase length	Distortion limit	Tuning algorithm	Bleu score	Bleu sub-scores	Runtime
4	6	MERT	14.75	54.0/21.1/10.2/5.4	2993
6	6	MERT	14.83	54.0/21.2/10.2/5.4	4327
8	6	MERT	15.05	53.2/21.0/10.3/5.5	4345
8	4	MERT	14.79	53.9/21.1/10.2/5.4	4520
8	2	MERT	15.06	53.4/21.0/10.3/5.5	2744
8	2	PRO	15.15	53.0/21.0/10.3/5.5	6218
6	4	PRO	15.24	53.5/21.3/10.4/5.6	7306

The processes used about 400 MB of memory during the tuning process. With four cores, there was about 40% utilization of each core at peak. They generally took between half an hour and an hour and a half (including other runs not shown here). The runtimes generally did not seem to have any correlation with the input phrase length, and any difference would probably be obscured by the machine utilization levels at runtime since the process takes so long. For

phrase length of four, the process converged after only 4 iterations, while it took 10 for the other two phrase length inputs. This probably is related to other convergence factors, not the given input.





Maximum Phrase Length

We first tuned with phrase length. For the three parameters for maximum phrase length that we tried, our BLEU score increased with each one. While in class we observed diminishing returns on higher phrase length around 5-6, our results do not confirm that. We actually had increasing gains. Of course, because of the small amount of input data that could easily be attributable to increased data or peculiar aspects of our input set and its interaction with the algorithms.

Distortion Limit

Next we tuned with with distortion limit. We found a strange dip at a limit of 4. With limits 2 and 6, the BLEU score was about the same at 15.06 and 15.06. However, it went down to 14.79 for distortion limit of 4. Also strangely, this went down even though the unigram precision went up, from 53.2 and 53.4 to 53.9 and that was the biggest change, while the other n-gram precisions only changed by .1. Since there are fewer n-gram overlaps for higher n, it makes sense that there is less room for variability; still one might expect the biggest changes from distortion to be in the higher n-gram range since distortions change how phrases move around. However, the changes in distortion could allow new phrases to be picked out of the phrase table, and seeing no other obvious changes in the translation output we attribute the slight differences to that.

Tuning Algorithm

We ran the PRO tuning algorithm on both the highest-scoring parameters we found (phrase length 8, distortion limit 2) and on the defaults, since ours were different from what would theoretically be expected (phrase length 6, distortion limit 4). PRO increased our BLEU score to 15.24 with maximum phrase length of 6, distortion limit 4, and to 15.16 with maximum phrase length of 8 and distortion limit 2 over MERT's 14.75 and 15.06. It took much longer to run--just over 2 hrs. The PRO paper describes the number of iteration as parameter to the algorithm ("Tuning as Ranking", Hopkins and May). PRO runs by default ran 26 iterations while the MERT rounds took around 10--until convergence. PRO took less time per iteration than MERT did, though that could easily be varied by changes the number of sentence pairs PRO samples. The more important scalability difference noted it the paper was the algorithm's ability to get accurate weights with more features and its generally improved results, not its speed. Indeed, the PRO algorithm does better than our previous best in all n-gram precisions as well as the merged final score. Its improvement of a few tenths for the merged score is consistent with the results in the paper.

Section 3: Output

We analyze the output of our best scoring tuning run, PRO with maximum phrase length of 6 and distortion limit of 4, in order to gain some insights into strengths and weaknesses of the system.

To start, the system is fairly good at common phrases and short sentences. [Below, reference translations are shown in bold and ours in italics]

at the same time , there will be partial roadblocks on many of the nation 's highways .
at the same time , at different points of countries , half of the road will be blocked as well .

In the above sample, the phrase “at the same time” is perfectly done, and the sense of road blockage throughout the country is generally conveyed, with some still accurate deviations from the references--nation vs country, road vs highways. However, some strange phrases come in: where did the idea of “half” come from? The addition and omission of certain parts of sentences is often evident. This probably comes from the strange phrase alignments seen in the phrase table. Perhaps with a lot more training data the likelihood of the strange alignments would be low enough that they wouldn’t appear in the sentences.

One obvious omission is numbers:
in new york , the dow jones industrial average jumped by 331 points , or 2.55 per cent , to close at 13,289.40 , while the broader-based s&p 500 index rose by 2.65 per cent .
a new york , the average industrial of increased points , or % vision to as the index the largest , the s&p 500 , took % .
Numbers are often completely omitted, especially uncommon ones. Numbers like 1 or 3 seem to be fine. It seems the decoding system should leave these the same in the translation--they probably haven’t been seen before. Yet for the ones it hasn’t seen, it simply leaves them out.

With regards to complex, lengthy sentences, they are usually not intelligible. Phrases might be correct, but ordering and changes of nouns from subject to object, etc, make it impossible to understand. Consider the following:

for that , they think that stock-market will have a rather lateral behaviour , non-free of volatility " so we do n't know the impact of the fed 's probability reduction types , for the high level of crude oil and the stimulating danger of the inflationary tensions " .
for this , they believe that the stock will behavior rather , not to " especially because we are not familiar with the impact of the likely lower rates of the fed , given the high level of crude and the danger to the stimulation of inflationary tensions "

What is “stimulating danger” in the reference becomes “danger to the stimulation.” Words are sometimes right, but meaning is garbled. Thus, a lot of the issues seem to stem from limited training data, but likely a better language model or some syntactic understanding would greatly help.

We end with a this translation:

the choice might break or make either man 's career , or it might make little difference .
this decision could eliminating or build the career of a man , but it has little difference .

The search clearly did not align whatever french phrase must literally translate to “eliminate or build” with the English “make or break.” It translated word-by-word instead of capturing the phrase.

Some consulted sources:

Hopkins and May. “Tuning and Ranking. ”Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1352–1362, Edinburgh, Scotland, UK, July 27–31, 2011.

Collins handout for Model1 and Model2.

Jurafsky and Martin textbook.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a Method for Automatic Evaluation of Machine Translation.” Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.

Lecture slides and lecture notes.

Moses documentation and tutorials. http://www.statmt.org/moses/.