Question 1: Annotators very frequently agree on the sentences alignments. Most word by word translation cases are very successful, but English and German have some important syntactical and morphological differences which can result in sentences (or parts of sentences) very different from one-another. Alignments tend to follow the diagonal of the matrix. This indicates that the syntax of the English-German pair of sentences is approximately the same. Then length of the sentence does not seem to influence the *sure* or *possible* alignments. Also when alignment is not diagonal, it means that one of the two languages has used a different syntax. This can be due to simple syntactical differences due to grammar (eg. German tends to put the verb at the end of the sentence, see sentence 111; in the present perfect tense, German divides the auxiliary from the main verb, see sentence 103), or transposition of pieces of sentences (see sentence 138).

Question 2: The less sentences are used, the lower the Precision and the higher the Recall. Conversely, the more sentences are used, the higher the Precision and the lower the Recall. The AER is not very much influenced by these changes, because it is an averaged measure of success of both Precision and Recall. To improve the AER score, both Precision and Recall need to go up at the same time. The Dice coefficient is a co-occurrence metric. The more sentences are used, the more the co-occurrences are correct because they work on more data, which results in a more robust Dice coefficient and Precision score. Conversely, with less sentences, the Dice coefficient works on more sparse and less exact data, which causes the Recall to improve by aligning almost anything to anything, and the Precision to decrease.

Score using 150 sentences:	**Score using 10000 sentences:**
Precision = 0.103819 Recall = 0.609937 AER = 0.830362	Precision = 0.149023 Recall = 0.378788 AER = 0.789858

Question 3 The higher the threshold, the higher the Precision and the lower the Recall. Conversely, the lower the threshold, the higher the Recall and the lower the Precision. This happens because by choosing a high threshold, there is a very narrow selection of co-occurrences, namely those words that have co-occurred over 90% of their global count. By lowering the threshold, the Recall goes up because also words with much fewer co-occurrences are aligned to each other. As for question 2, the AER goes up and down by a few points as the threshold gets changed, but never dramatically. In fact, AER would go down considerably only if both Precision and Recall go up at the same time.

Score using 0.9 threshold:	**Score using 0.1 threshold:**
Precision = 0.421569 Recall = 0.119803 AER = 0.812808	Precision = 0.068368 Recall = 0.829105 AER = 0.881665

Question 4: The Dice coefficient is a similarity metric that is calculated using number of co-occurrences as a basis, so it is not a proper alignment exercise. This results in some non-sense alignments, eg. the end-of-sentence period is aligned to various words, due to its very frequent co-occurrence with almost any word in any sentence. The same happens with many stop-words, like “the”, “and”, “in”, “of”, “to”, “for”, etc. By varying the threshold to higher values, most non-sense stop-words alignments are filtered out. So this causes less random and more precise alignments. These tend to be either word-to-word exact translations (eg. days of the week, numbers or proper nouns), or very infrequent words, which return a high Dice coefficient due to their almost exclusive occurrence in one or just a few sentences.

The IBM Model 1 is implemented in Python, run over 5 iterations gives the following results after scoring alignments

Precision = 0.490107 Recall = 0.536293 AER = 0.488486

Please note: By making a minor change in last sentence of code, this file instead of outputting the alignments, outputs the t_ef values, which is used in IBM model 2 in Q9. This t_ef_ibm1 file is submitted along and needs to be present in same directory before running mymodel.py

Question 5 As from fig5-1, iteration =12 and some data = first 1000 sentences of corpora, the plot shows log likelihood suddenly dipping at iteration 2 then increasing suddenly in 3rd and 4th iterations, and then smoothly converging around 8th iteration onwards. It does appear to converge and becomes an almost straight line after iteration 12. Comparing with AER after each iteration, log likelihood and AER show a rather peculiar relation in graph. AER decreases with increasing data log likelihood linearly from 4th to 10th iteration. Between 1st and 3rd iteration, the AER values seem to be unstable.

Question 6 Run over a sample of **10000** sentences in corpora, and choosing frequent and infrequent words as "**the**" and "**aegis**" whose frequency in corpora is 80387 and 12 respectively, their translation distributions are shown in fig6a-1 and fig6a-2 at the end of report. For both, x axis is the set of foreign words and y axis is the t_ef value (translation probability). Word **aegis** tends to align with a lot of words, sometimes in a spurious manner, whereas the word **the** tends to align strongly with atleast 3 german words and with atleast 10 german words with t_ef value between 0.15 – 0.20. The shape of distribution(for both words) shows a pattern of regularly aligning as one goes over the German corpora.

Question 7 Run over a sample of **10000** sentences in corpora**,** and choosing morphological variants "**rise**" and "**rose**" whose frequency in corpora is 2236 and 176 respectively, their translation distributions are shown in fig7-1 and fig7-2 at the end of report. For both, x axis is the set of foreign words and y axis is the t_ef value (translation probability). They are similar even though their frequency in corpora has a huge difference. This is unexpected, as the translation prob of **rise** was expected to be much more. Adding to the fact that the word **rose** is not only the morphological variant of verb (rise and rose) but can also be the noun **rose.** For both words, the translation prob is very high (0.4-0.45) compared to a few foreign words (likely to depict the true meaning in german), but for all remaining words in corpora, this probability remains between 0.0-0.15

Question 8 For stop-words, like "the", "and", "in", "of", "to", "for", Model1 alignments are still the cause of randomness in alignments and perform poorer when compared to human alignments. In general, German tends to use a few less words than English to express the same meaning. When this happens, more than one English word need to be aligned to one German word (eg. sentence 4, Eng *"… on behalf of all the…"*). In such cases, Model1 performs poorly because it does not talks the null value assumption, while the human alignments tend to perform far better due to better intuition. In cases when one of the two languages uses an expression that is specific to that language, but does not exist in the other (eg. sentence 5, Eng *"Please rise…"* translates Ger *"Ich bitte Sie…"*), Model1 again differs from human alignments. The use of idioms (eg. sentence 137, Eng *"There is no room for…"*). Use of the particle *"to"* in English (eg sentences 20, 30, 49, 55, 56, 57, 58, 69, 71…..) causes the aligner to align to multiple german words, but for this case, Model1 perform better than Dice aligner and more regularity to the human alignments is observed. Same applies to phrasal verbs in English (eg. sentence 15, Eng *"… come up…"*).

Question 9 A better probabilistic model of alignment is IBM Model2. After working on Model1, Model2 seems a natural choice, though the code runs slower. Model1 code is modified to print the translation prob (t_ef) to a temp file called "t_ef_ibm1" which this new model 2 takes as input. This extra file is attached with the submissions with the name t_ef_ibm1 **(please note that this file has translation prob of 5000 sentences only and not the full corpora).**

The score for this new model **(i=5, n =5000)** - is

```
Precision = 0.455403
Recall = 0.496124
AER = 0.525723
```

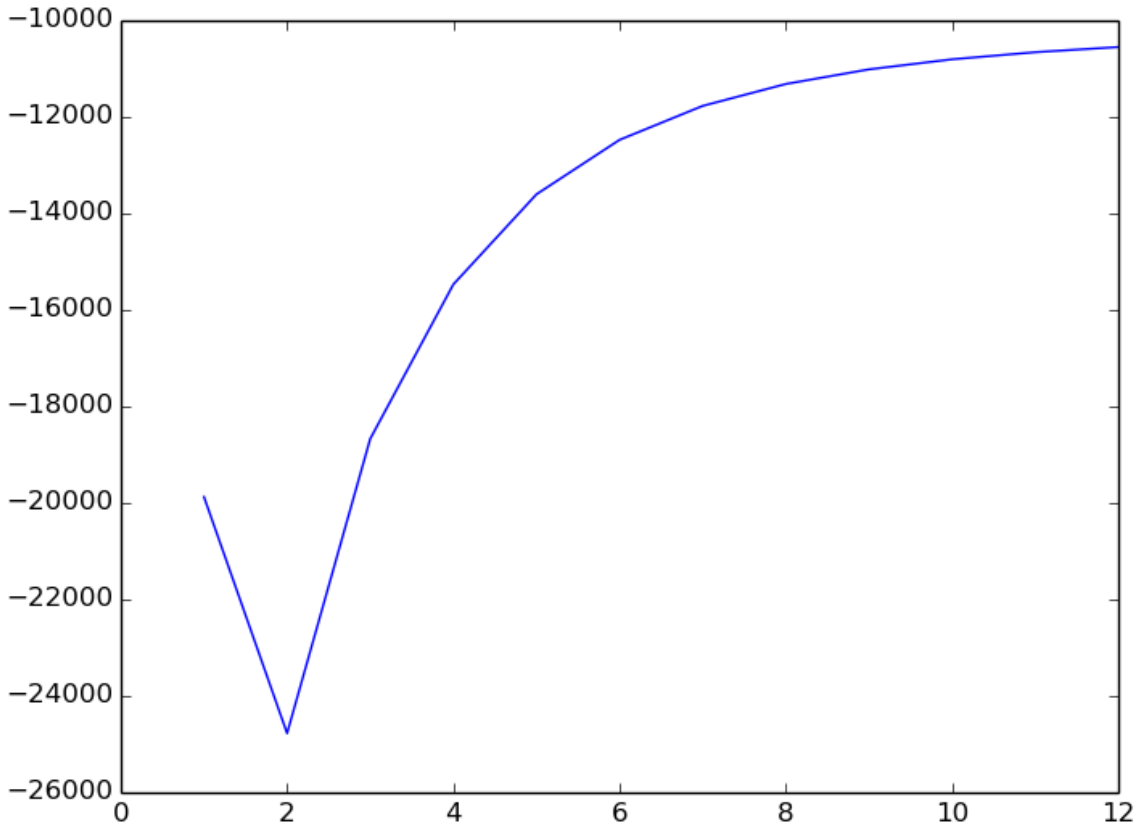
−10000
−12000
−14000
−16000
−18000
−20000
−22000
−24000
−26000
0
2
4
6
8
10
12

Fig5 1

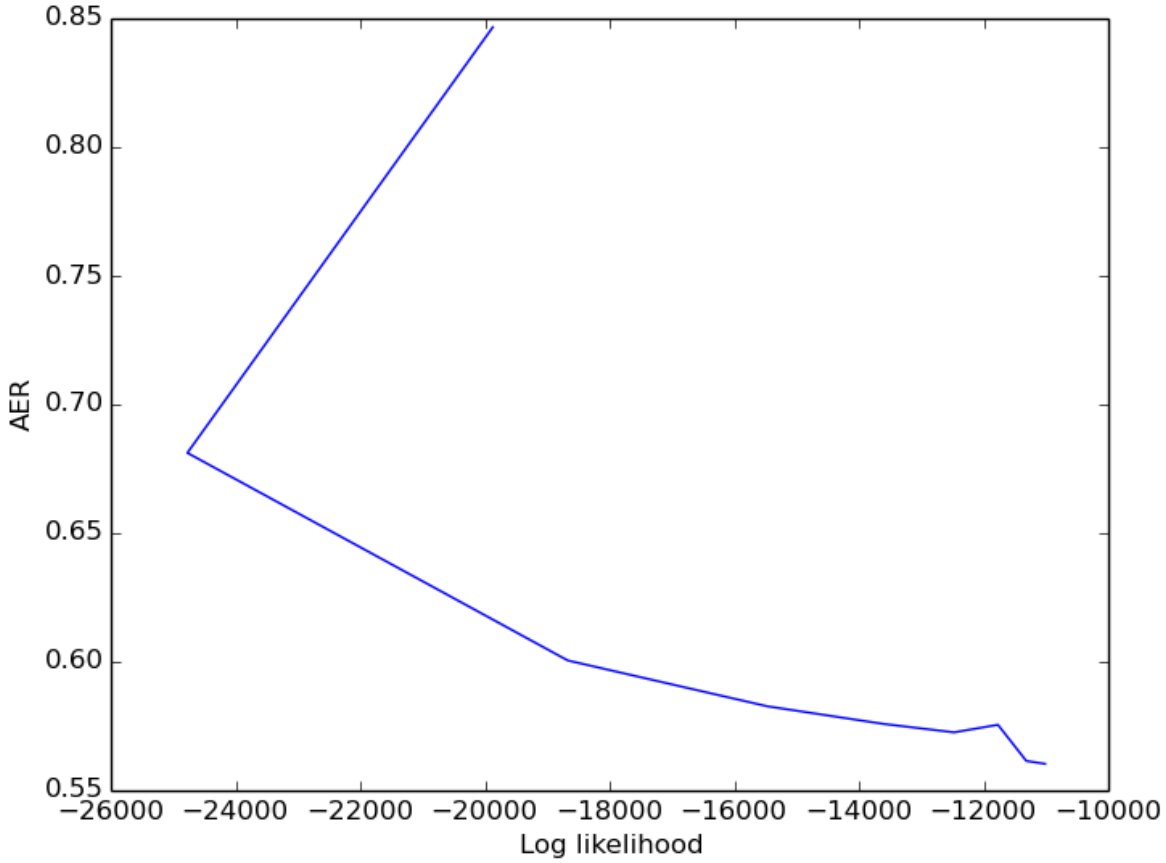
0.85
0.80
0.75
0.70
0.65
0.60
0.55
AER
−26000
−24000
−22000
−20000
−18000
−16000
−14000
−12000
−10000
Log likelihood

Fig5 2

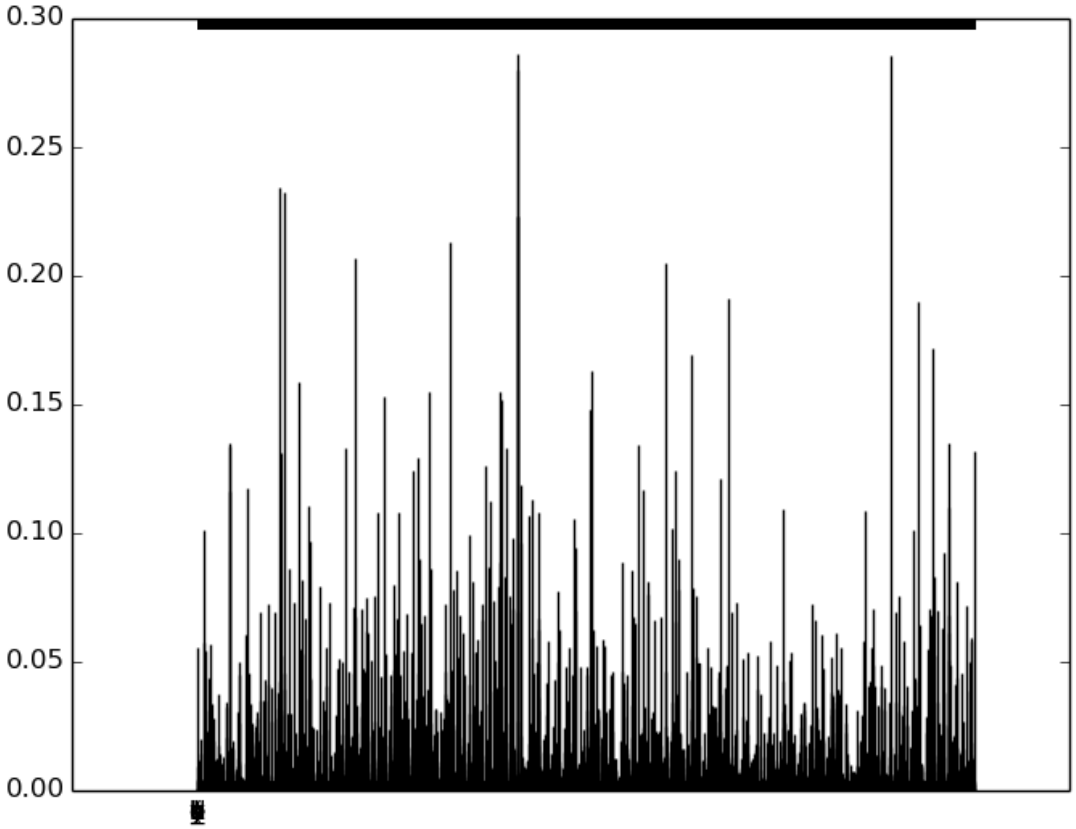

Fig6-a 1

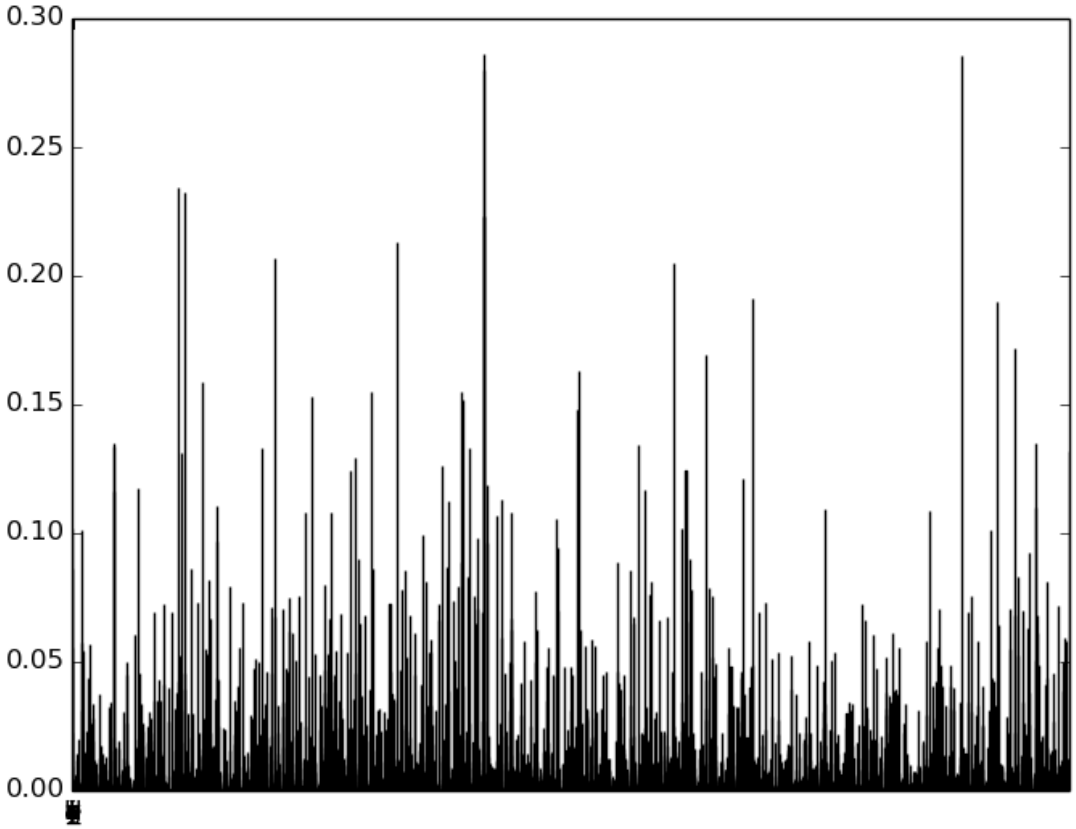

Fig6-a 2

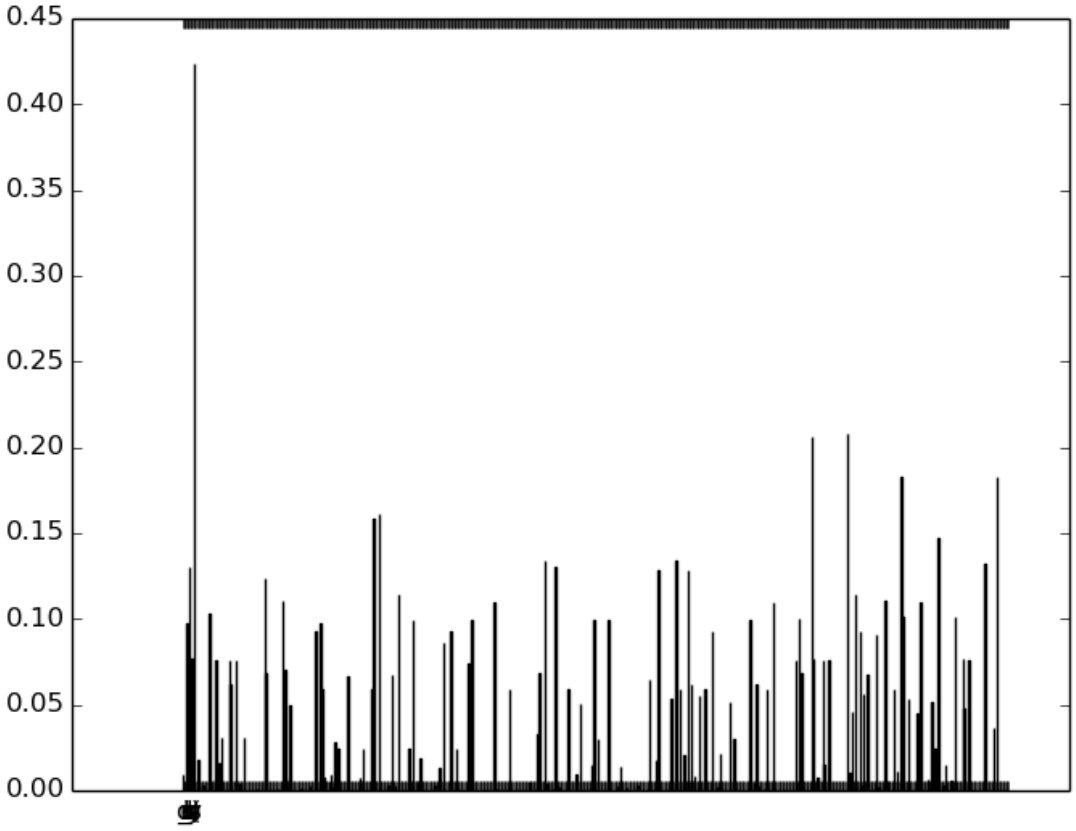

Fig7 1

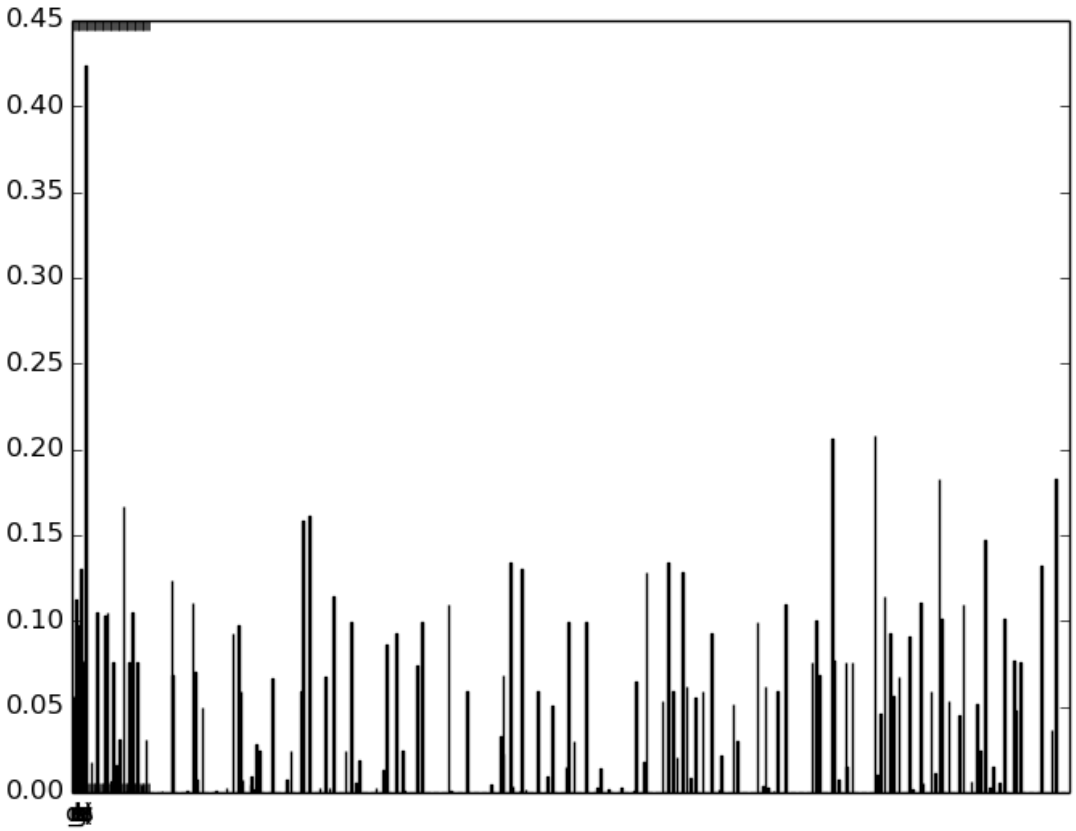

Fig7 2