Project: IBM SMT Model 1 and 2 Word Alignment

Markos Mylonakis
m.mylonakis@uva.nl

1 Introduction

Statistical Machine Translation (SMT) models define a way to explain how sentences get translated. When translating sentences from a source language to a target language, we use the *Noisy Channel* paradigm to consider each source language sentence $\mathbf{f}$ as a *corrupted version* of an initial target language sentence $\mathbf{e}$. To formalize this one can use Bayes' law to rewrite the task of finding the most probable target language translation given the source language at hand, as finding the most probable source sentence that could have generated $\mathbf{f}$ as its corrupted version.

$$\hat{e} = argmax_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) = argmax_{\mathbf{e}} P(\mathbf{f}|\mathbf{e}) P(\mathbf{e}) \tag{1}$$

This splits the problem into defining a *language model* $P(\mathbf{e})$ over the source language and a *translation model* $P(\mathbf{f}|\mathbf{e})$ between sentences of the source and the target language. The latter is where the focus of most SMT research efforts falls upon.

For the translation model one can assume that *word alignments* $\mathbf{a}$ exist between the words of the source and the target sentence. These alignments can be intuitively understood as links between single words of the source and target sentence, which indicate that the linked words are translations of each other. In the general case, a single source or target word can have alignments to more than one word on the opposite side (e.g. think of 'not' being aligned with both 'ne' and 'pas' when we deal with English and French). An example alignment between an English sentence and its French translation can be seen in figure 1.

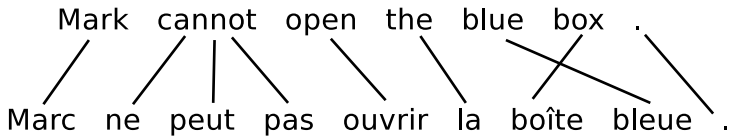

Figure 1: Alignment between English and French. Notice how a word ('cannot') has more than a single word alignment. In addition, the order of the aligned words is not always of course the same (as it happens here with 'blue' and 'box' and their translations).

By introducing the alignments, one can reformulate:

$$P(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e}) \tag{2}$$

A fundamental problem in SMT is then word aligning a bitext. In this case we do not seek for the most probable translation $\mathbf{e}$ given $\mathbf{f}$ because we have access to both. What we do care about is word aligning them, that is find the most probable alignment $\mathbf{a}$ between their words. Word aligning a bitext is frequently the first step for training many modern SMT systems.

The *IBM Model 1* for SMT defines a simple way to explain how sentences get translated. The parametrization of the model involves estimating *translation probabilities* of the type $P(f|e)$ for single words f from the source language and e from the target. Getting a *Maximum Likelihood* estimate of the parameters from a bitext involves the usage of the Expectation-Maximisation algorithm. A nice property of IBM Model 1 is that there is only a single maximum of the likelihood function. This means that initialisation is not important (we can start with uniform).

IBM Model 2 builds up from Model 1 by adding *alignment probabilities*. These are based solely on the word positions i, j that we are aligning together and the lengths l, m of the two sentences. They are thus of the form $P(i|j, l, m)$. What we would like these probabilities to signal is that, for example, the 3rd word of a French sentence is more probable to be aligned to the 2nd, 3rd or 4th word of the English sentence than to the 30th. It sounds (and is) simplistic, but still it is much better than Model 1, which is completely ignorant of the distance between aligned words. Model 2's likelihood function is not guaranteed to have a single maximum, so initialisation

is important. What we usually do then, is to initialise Model 2's translation probabilities by the estimates of Model 1 for the same parameters.

A useful feature that both models enjoy is that we can perform EM without the need for approximating any of its two steps. Things get more tricky when we move to IBM Models 3 and above, and approximative techniques must be used to run EM using them. That is why, in this project, we will constrain ourselves on Model 1 and 2.

2 Deliverables

What you need to do is:

- Write an implementation of the EM algorithm to train both IBM Model 1 and 2 parameters from a bitext training corpus. As forementioned, you may use the estimates from Model 1 to initialise the EM estimation using Model 2.
- Write a program that then word aligns a test bitext using both IBM Model 1 and 2.
- You will be given a small bitext. Use the word alignment program you developed to word align it using the two models and with parameter estimates you got for them by using EM. Submit the word aligned text in GIZA format.
- You will be given a manually aligned reference version of the same bitext. Evaluate the word alignment you derived using your program and the two model estimates against it and report precision, recall and F-1 score.
- Write a report following the relevant guidelines that will be given to you, including your empirical results. Some interesting points: Go through the basic intuitions behind SMT, describe the general framework that SMT models use and address IBM Model 1 and 2 in particular. Discuss training the models and include equations defining how you performed the word alignment with them. In the discussion/conclusion section also address the shortcomings of each model. Is Model 2 doing better than its simpler counterpart? If yes why (show some examples)? Shortly comment on what kind of relations between

the translated sentences pair could be used by more complicated models to increase performance.

- Submit your report, all programs and scripts you have developed and the GIZA format alignment using the relevant guidelines. Please choose one of the following computer languages for the implementation: C, C++, Java, Perl, Python, PHP. You will receive instructions on how to submit your work later on.

3 What will you learn?

Through this project, you will gain hands on experience on estimating parameters for some of the IBM Word Alignment Models. These models are used by most of the SMT researchers up to this day to perform word alignment. It is important to note that we will use the Expectation Maximisation algorithm to do this, which is something with a value of its own. We will then move one step further and actually construct a word aligner, and use it to align unknown text. Finally, we will critically examine the resulting alignments and brainstorm on what can be done to do even better. This last step will hopefully provide some links to the more complicated IBM Models.

4 Team Organisation and Skills

This is a project for a *team of three* highly capable students. Some skills that will be needed are: Familiarity with the concept of mathematical optimisation (finding the maximum of a function given constraints), understanding how the EM algorithm works, programming skills for the implementations, critically examining alignment results. An ideal team would combine team members that can help each other in the above challenges.

5 Support

This is a challenging project. Moreover, to understand what you are doing, it includes going through some material which will probably not be covered in detail in the course lectures. For this reason, a meeting with the team members will be arranged shortly after the project kickoff, to discuss these

additional topics and discuss all the tasks that you will perform. After that, we will arrange subsequent meetings as needed, as the project moves on. If you want to ask more questions about the project before commiting to it, please feel free to e-mail me.

6 Readings

[1] is a good primer on the intuitions and the foundations of SMT and the IBM Models. It does not include many details though. All IBM Models are explained in detail in [2]. You may find everything that you need about IBM Model 1 and 2 there. Of course, you can further make use of other papers on the subject that you might find (remember to cite them!).

References

[1] P. Brown, J. Cocke, S. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. A statistical approach to language translation. In *COLING-88*, 1988.

[2] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263 – 311, June 1993.