Language and Speech Processing

Statistical Machine Translator Aligner

Sanne Korzec - 0252700

Nicholas Piël - 0104612

Bart J. Buter - 9910522

{skorzec,npiel,bjbuter}@science.uva.nl

1 Introduction

At the end of the 19th century, L. L. Zamenhof proposed Esperanto; it was intended as a global language to be spoken and understood by everyone. The inventor was hoping that a common language could resolve global problems that lead to conflict. This idealistic idea did not reach its full potential, yet there are still scientific fields pursuing its legacy, though not necessarily from an ideological point of view. The Internet contains billions of web pages, as they come in all kinds of languages, a great deal of information is not available to us. A practical application would be a browser that translates these pages in a preferred language. In the field of statistical machine translation (SMT), we try to build algorithms to translate from one language to the other by mere statistics taken from large bi-text corpora. When we adopt the SMT approach, we represent all individual or groups of words (cepts) as having a connection to zero, one or many foreign cepts under a probability value. This means that both the alignments and the probabilities need to be extracted from the bi-text. The main problem here is that we need the alignments to estimate the probabilities and the probabilities to estimate the alignments. These kinds of problems can be solved with the EM algorithm, one approach [2, 3] is particularly favored due to its estimation of reasonable models.

In this paper, we will present the SMT Aligner (SMTA), which word aligns French-English sentences given word translation probabilities provided by [1] to estimate good alignments under the assumptions of IBM model I. We will then compare results from alignments that use the null word with ones that don't. We will also introduce a heuristic to increase the F1-score results by 4 percent.

In section 2 we describe the theory which forms the basis of the SMTA. Section 3 is used to describe the research methodology. In Section 4 we present our results and we conclude section 5 with the discussion.

2 Theoretical Model

Consider the English sentence e_1^4: "it is quite understandable." and the French sentence f_1^6: "ce est tout à fait compréhensible.". The French sentence needs six words to represent four English words. But one can connect the two sets of words in many different ways. Our task is to find an optimal alignment. We will use an SMT model to statistically choose the most likely alignment. In this section we will describe the theory which is the basis for SMT and the SMTA. We will describe the language and the translation model, the noisy channel model, the used heuristics to extract the alignments and the related theories and concepts. These theories are based on [2, 3, 4, 5, 6].

2.1 Statistical Machine Translator as noisy channel decoder

In SMT we take the view that every French string f, is a possible translation of e. We assign a probability $P(f|e)$ to each pair of f, e. This model is based on the notion of a noisy channel. In a noisy channel, a message $\bar{M}$ is sent through a channel, on the other side of the channel the message M is received. The received message M is not necessarily the sent message $\bar{M}$. It is the task of the decoder to try to make an estimate $\hat{M}$ of the original message $\bar{M}$ from the received message M.

We can use the noisy channel model as follows: the original English string being sent is a sequence of words $e_1^{\bar{m}}$, the received message is a sequence of french words f_1^m. The Statistical Machine Translator is the decoder which creates an estimate e_1^m of the original English sentence given f_1^m. Formally, we can define this as finding the sequence e_1^m that maximizes $P(e_1^m|f_1^m)$.

2.2 The Language and Translation Model

Using Bayes on the Statistical Machine Translator decoding task we get:

$$\underset{e}{argmax}\ P(e_1^m|f_1^m) = \underset{e}{argmax}\frac{P(f_1^m|e_1^m)P(e_1^m)}{P(f_1^m)}$$

The term $P(f_1^m)$ is equal for all sequences. Therefore we can leave this out of the equation resulting in the fundamental decomposition:

$$\underset{e}{argmax}\ P(f_1^m|e_1^m)P(e_1^m)$$

The model can be split into two parts: the language model which describes the probabilities of a word sequence i.e. $P(e_1, e_2, ..., e_m) = P(e_1^m)$, and a translation model describing the probabilities of a word sequence f_1^m given a word sequence e_1^m i.e. $P(f_1^m|e_1^m)$. The noisy channel model indicates that there are two processes at work to ultimately produce the received message M. Namely the process that produces the message $\bar{M}$ and the process of transmitting $\bar{M}$ through the channel $\bar{M} \rightarrow M$.

These two processes are modeled in the language and translation model. We assume that the English word sequence e_1^m is generated from a probabilistic process, the related model is the language model $P(e_1^m)$. We will not elaborate on this model as it can be found in many papers [7, 8]. The channel is also assumed to be a probabilistic process which is modeled by the translation model $P(f_1^m|e_1^m)$.

2.3 IBM model 1

Brown et al. [2] proposed an incremental translation model. Their IBM models (1-5) use the EM algorithm to estimate the translation probabilities between words. IBM Model 1 is the most simple of the proposed translation models, one of its goals is to provide initial estimates for the other IBM models, but it has also been used for several other methods [10]. We will not go into the details of estimating the translation probabilities, but we do need to address some of the formulas and assumptions IBM model I uses. First, we assume that an alignment is a collection of connections, where each french word has one connection. Given all possible alignments we can state that:

$$P(f,a|e) \quad = \quad P(m|e)\prod_{j=1}^{m} P(a_j|a_1^{j-1},f_1^{j-1},m,e)P(f_j|a_1^{j},f_1^{j-1},m,e)$$

We assume that $\prod_{j=1}^{m} P(a_j|a_1^{j-1},f_1^{j-1},m,e) = \frac{1}{l+1}$ thus, this only depends on l, the length of the English string. Therefore, we can write:

$$P(f,a|e) \quad = \quad P(m|e)\frac{1}{(l+1)^m}\prod_{j=1}^{m} P(f_j|a_1^{j},f_1^{j-1},m,e).$$

Next, we assume that the translation probability $P(f_j|a_1^{j},f_1^{j-1},m,e) \quad := \quad t(f_j|e_{a_j})$ only depends on f_j and e_{a_j}. Now we get:

$$P(f,a|e) \quad = \quad P(m|e)\frac{1}{(l+1)^m}\prod_{j=1}^{m} t(f_j|e_{a_j}).$$

Last we assume $P(m|e)$ to be a constant ϵ. Thus we arrive at the formula:

$$P(f,a|e) \quad = \quad \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m} t(f_j|e_{a_j}),$$

summing over all alignments we get $P(f|e)$:

$$\begin{aligned} P(f|e) \quad &= \quad \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l} t(f_j|e_i) \\ &= \quad \sum_{a} P(f,a|e) \end{aligned}$$

We see that we have arrived at a model where the order of the words in the sentence is not important for the probability $P(f|e)$. We can also see that if our m and l are constant the whole first term in the formula becomes a constant, Thus:

$$\frac{\epsilon}{(l+1)^m} \quad \Rightarrow \quad c$$

2.4 Alignments

Under IBM model I, we assume that each target word is to be generated by exactly one source word, which can also be the null word. This means that an alignment a, which aligns a sequence of source words to a sequence of target words, can be represented as vector a_1^m, where each a_j is the word position of the source sentence word generating target word f_j. We can now use IBM Model I to represent the most likely alignment as: $a = \underset{a}{argmax}\prod_{j=1}^{m} t(f_j|e_{a_j})$. As we have no dependencies, this is equivalent to

$$a = \prod_{j=1}^{m} \underset{a}{argmax}\ t(f_j|e_{a_j}) \qquad \text{(def. 1)},$$

making the task of finding a_{max} less demanding computationally.

2.5 Optimizing the heuristics

We implemented two heuristics: Greedy, the default model and NSB. Both heuristics are based on the assumption that the word order of French and English is similar [3]. The Greedy approach exploits the formula given in definition 1, but it can sometimes produce errors when confronted with sentences in which the same words appear more than once. Model 1 assigns a connection based on the same probability, resulting in one word receiving all connections, depending on the order in which the sentence is traversed in the algorithm. Clearly, we do not want this, as it is intuitively highly unlikely that such a translation from all ambiguous words to one word exists. The NSB heuristic therefore keeps track of the number of times a word is connected and prefers words that haven't been connected yet.

3 Research Methodology

We implemented the SMTA based on the models and principles described in section 2. We used a subset of the Canadian Hansards bilingual corpus [3]. The word translation file was provided by [1]. The test set was 447 sentence pairs, with variable lengths between 2 and 30. These were manually labeled into two gold sets, one set allowed inclusion of the null word, in the other set they were omitted if no satisfactory alignment could be found. Alignments were labeled as "Sure" or "Possible". We then define the sets $S = "Sure"$ and $P = "Possible" \cup "Sure"$ [9]. The SMTA generates an alignment A. We evaluate our alignment A by comparing to the sets, using the following measures:

$$Recall = \frac{|A \cap S|}{|S|}$$

$$Precision = \frac{|A \cap P|}{|A|}$$

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

To generate the ROC we introduced a threshold which was compared to the probabilities that constitute A. For the non null alignment set, alignments under the threshold

were omitted from A and for the the null alignment set, alignments under the threshold were assigned to the null word.

4 Results

We have tested our SMTA with two different heuristics, the greedy and the nsb heuristic on two different data sets. One with NULL values and one without NULL values. The performance can be seen in Table 4. We will discuss these results in more detail in the coming sections.

	Recall	Precision	F1
NSB with NULL	**0.87**	**0.74**	**0.80**
Greedy with NULL	0.85	0.71	0.77
NSB without NULL	0.87	0.69	0.77
Greedy without NULL	0.85	0.68	0.75

Table 1: SMTA Results

4.1 With NULL vs Without NULL

In table 4 we can see that by making use of the NULL value the recall stays the same but the precision increases. This is because when one wants to align two sentences it is very well possible that some words are impossible to align. For example: “le ministre chargé de les transports.” versus “minister of transport.” In this case the French word ’chargé’ has no equivalent in English. In the alignment without NULL values, the total score possible for recall is the returned amount of the other 6 words (including the dot), for precision the total score possible would be 4/4. However, if we make use of the NULL value this will mean that the maximum recall would be 7 out of 7 and for precision 5 out of 5.

4.2 Different Heuristics

We have implemented and tested two kinds of heuristics, the greedy heuristic and the NSB heuristic. Our NSB heuristic outperforms the greedy heuristic in both recall and precision (Table 4). This is expected, since it is specially targeted to classify a subset of sentences which tend to be misclassified by the greedy heuristic, namely those with recurring words. For example when we take a look at the French sentence. "Oh, oh!" and its identical English counterpart. We can see the generated alignments in Figure 1. Greedy aligns the first ’Oh’ with both French ’Oh’s where the NSB heuristic make the correct alignment.

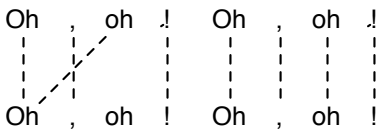

Figure 1: “oh, oh!” Alignment. Left: Greedy. Right: NSB.

4.3 Performance Over Sentence Lengths

The problem of alignment becomes increasingly difficult with the length of the sentence as the possibilities of alignment grows exponentially. It is therefore interesting to look at how the performance varies over various sentence lengths.

In Figure 2 we can see the performance of the greedy heuristic and in Figure 3 we can see the performance of the NSB heuristic. In both Figures we can indeed see this decline for longer sentence lengths.

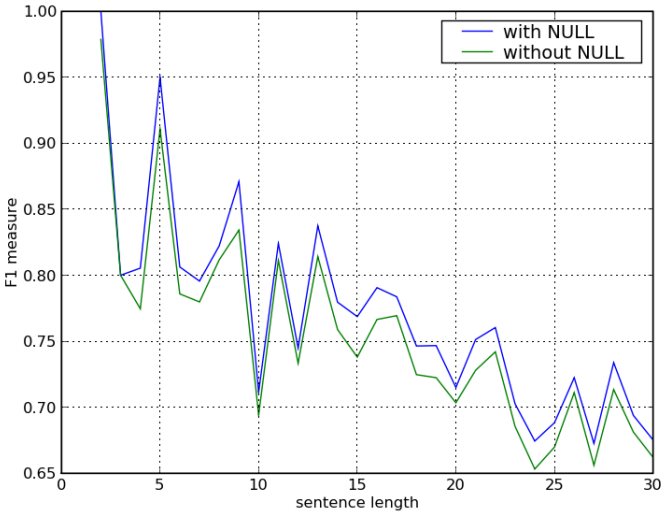

Figure 2: Greedy F1-score over sentence lengths.

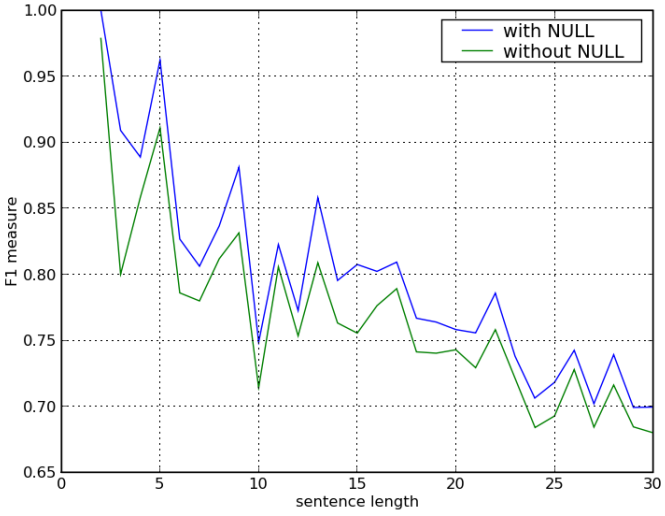

Figure 3: NSB F1-score over sentence lengths.

4.4 Optimizing Threshold

We have tried to find a sweet spot by plotting a ROC over various thresholds. It can be seen in the Figure 4 that not only the NSB heuristic has a higher score then the greedy heuristic but also that the threshold can be used as a nice way to tweak the performance to adjust for a preference over precision vs recall. As increasing the threshold will increase precision but at a decreasing performance in recall.

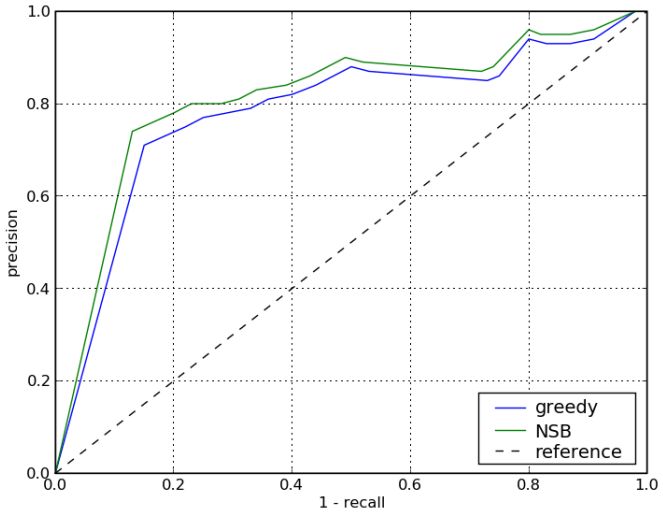

Figure 4: ROC Curves for greedy and NSB alignments with the NULL word allowed. We interpolated to the points (0,0) and (1,1).

5 Discussion

In this paper we have shown to be able to achieve an F1 measure of 0.80, this number seems to be able to compete with the results from [10]. We obtained these results by boosting the already high performance of our greedy approach with another 4%. This was done by targeting just one of model 1's weaknesses, namely its position independence. While this approach shows a boosted performance its has its limitations. The NSB heuristic assumes that recurring words in the source language point to the identical ordered recurring words in the target sentence. This does not have to be the case and could possibly lead to degrading performance over some sentences.

We have also shown that the threshold can be used to adjust the results to a preference for precision or recall. However, this does not increase the F1 measure. Another point that deserves attention is the fact that we calculate our precision and recall from different sets which is beneficial to our results. Even though this seems intuitively incorrect, many experts use the same evaluation measure [9, 10].

IBM model 1 has some widely known structural limitations such as only supporting a many-to-one and not a many-to-many alignment. It completely ignores the positions of the words in the sentences and any other alignments already set. But by taking these shortcomings into account it is still able to achieve good results. When expanding to a more complicated model, it would seem beneficial to allow many to many connections in the alignment models to improve performance. This is what is being done in IBM Models 2,3,4 and 5. However, we make the problem computationally intensive this way. Another approach would be to take the positions of the words into account and making the foreign sentence length dependent on the given sentence words.

5.1 Future work

More complex models that can deal with many to many relations create tough problems with respect to the complexity. Neuro evolutionary algorithms specialize in finding a suboptimal solution in very complex problems. Many papers report good results in the field of reinforcement learning [11]. Aligning cepts is also a discrete complex problem. We feel that the SMT domain could very well be combined with neuro evolutionary algorithms. A problem with the more complex IBM models is that we need to make a lot of assumptions, that do not necessarily hold. When we use neuro evolutionary networks we can benefit as we do not need to make these assumptions when creating a model, which will probably results in a higher performance.

References

[1] M. Mylonakis, http://staff.science.uva.nl/~mmylonak/index.html, 2007.

[2] PF. Brown, VJ. Della Pietra, SA. Della Pietra, and RL. Mercer. The mathematics of statistical machine translation: parameter estimation. Computational Linguistics, 19(2): 263 - 311, 1993.

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

[4] Manning, Schutze. "Foundations of Statistical Natural Language Processing". The MIT Press, 1999.

[5] Jurafsky, Martin. "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition". Prentice-Hall, 2000.

[6] K. Sima'an. Language and speech Processing course 2007. http://staff.science.uva.nl/~simaan/D-LangAndSpeech03/LangANDSpeech07.html, 2007.

[7] N. Piël, B. Buter, S. Korzec. Language and Speech processing. Course report, University of Amsterdam, 2007.

[8] G. Maltese, F. Mancini. An automatic technique to include grammatical and morphological information in a trigram-based statistical language model. In Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing, 1992.

[9] FJ. Och, H. Ney. A systematic comparison of vairous statistical alignment models. Computational Linguistics, 29(1):19-51, 2003.

[10] RC. Moore. Improving IBM Word-Alignment Model 1. From Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp. 519-526. 2004.

[11] KO. Stanley, R. Miikkulainen. Evolving Neural Networks through Augmenting Topologies. The MIT press journal. 2001.