Statistical Structure in Language Processing
IBM Model 1

Cristina Gârbacea
10407936
cr1st1na.garbacea@gmail.com

Sara Veldhoen
10545298
sara.veldhoen@student.uva.nl

Abstract

In this paper we present the theoretical notions underlying the IBM Model 1 we implemented and the Expectation Maximization algorithm we applied to train the given parallel Dutch - English corpus. We also came up with our own extension to the model, which we compare and evaluate against the basic IBM model 1.

1 Introduction

The goal of finding the best English translation $\mathbf{e}$ of a foreign sentence $\mathbf{f}$ is modeled under the assumption of the *noisy channel* hypothesis, which considers the foreign sentence as a "corrupted" instance of the original English sentence. The English sentence which is at the source of its foreign counterpart is an unknown issue, and thus the task of translating $\mathbf{f}$ into $\mathbf{e}$ becomes one of maximizing the probability of the English sentence given the foreign sentence, $P(\mathbf{e}|\mathbf{f})$. According to Bayes' theorem, the translation problem can be expressed as:

$$P(\mathbf{e}|\mathbf{f}) = arg\ \max_{\mathbf{e}}\ P(\mathbf{e}) \cdot P(\mathbf{f}|\mathbf{e}) \qquad (1)$$

where $P(\mathbf{e})$ denotes the language model, that takes care of the fluency of the output, and $P(\mathbf{e}|\mathbf{f})$ denotes the translation model, that makes sure the translation is adequate.

In order to be able to approximate this probability, we need to make assumptions. In (1), (2) a series of increasingly complex models is presented. We implemented the first model, hereafter IBM model 1.

In what follows we present this model in section 2, together with the Expectation Maximization formula used for training this model. We came up with an improvement over this model, which we introduce in section 3. Then comes an evaluation of the alignment quality for both models in section 4 and finally we conclude in Section 5.

2 IBM Model 1

The assumption underlying the IBM models is that translating text comes down to aligning the words of the foreign and the English sentence, and translating the words independently. The words of the English sentence $\mathbf{e}$ are called *cepts* and generally within the IBM models groups of words in the foreign sentence $\mathbf{f}$ cannot be aligned to groups of cepts. Rather, each foreign word is aligned to one cept or, in case there is no English correspondent, to the so called *null* word. Because of this simplification, an alignment can be represented as a vector $\mathbf{a}$ that has the same length as the foreign sentence: m.

The translation probability $P(\mathbf{f}|\mathbf{e})$ can be rewritten as the sum over all possible alignments $\mathbf{a}$ of conditional probabilities $P(\mathbf{f}, \mathbf{a}|\mathbf{e})$:

$$\begin{aligned} P(\mathbf{f}|\mathbf{e}) &= \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e}) \\ &= \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}, m|\mathbf{e}) \\ &= \sum_{\mathbf{a}} P(m|\mathbf{e}) \times P(\mathbf{f}|\mathbf{a}, m, \mathbf{e}) \\ &= P(m|\mathbf{e}) \sum_{\mathbf{a}} P(\mathbf{f}|\mathbf{a}, m, \mathbf{e}) \qquad (2) \end{aligned}$$

$$\begin{aligned} P(\mathbf{f}|\mathbf{a}, m, \mathbf{e}) = &\prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \\ &\times P(f_j|a_1^{j}, f_1^{j-1}, m, \mathbf{e}) \qquad (3) \end{aligned}$$

EM Training The input to the training is a corpus of paired sentences. To estimate the latent variables, alignments, a version of expectation maximization is used. Training focuses on the parameters in equation 3. The first term, the alignment probability, is assumed to be uniform in IBM Model 1.

The initial step of EM consists in setting uniform translation probabilities $t(e|f)$ over the target vocabulary.

In the E-step, the alignment probabilities are computed. Note that this is done implicitly, in the process of gathering the fractional counts:

$$c(f|e;\mathbf{f},\mathbf{e}) = \frac{t(f|e)}{t(f|e_0)+\ldots+t(f|e_l)} \times \sum_{j=1}^{m}\delta(f,f_j)\sum_{i=0}^{l}\delta(e,e_i) \quad (4)$$

The translation probabilities are re-estimated in the M-Step, using the obtained counts:

$$t(f|e) = \lambda_e^{-1}\sum_{s=1}^{S} c(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \quad (5)$$

$$\lambda_e = \sum_{f}\sum_{s=1}^{S} c(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}) \quad (6)$$

Where equation 6 (λ_e) defines the normalization parameter to get proper probabilities.

3 Improvement Over IBM Model 1

In IBM Model 1, all alignments are equally probable. The Viterbi alignment can thus be found by simply choosing, for each word, the cept that has the highest translation probability.

An improvement can be made by introducing a probability function over alignments: the first term in equation 3. For simplification, we assume independence of each alignment decision, and independence of lexical items, such that we can compute $p(a_j|j,m,l)$ for each word.

This is also done in IBM Model 2. However, instead of training the alignment probabilities, as in IBM Model 2, we use a heuristic function. We base this function on the assumption that a word in the input should stay close, i.e. we want to favor $a_j = j$ over $a_j > j$ and $a_j < j$. Furthermore, we tried to favor choosing the null word as presented in (3), but that resulted in a drop in performance. The alignment function we created is defined as

$$p(a_j|j,m) = \frac{1}{m+|a_j-j|} \quad (7)$$

4 Experiments and Results

We implemented IBM Model 1 and trained it with 20 iterations of the Expectation Maximization algorithm on the given Dutch-English parallel corpus consisting of 1.000 sentences. We evaluated the alignment quality against the quality of the alignments provided by Giza++. Below we report the results we obtained, as well as the results of our approach:

	Precison	**Recall**	**F1-score**
IBM Model 1	0.9111	0.9053	0.9063
Our extension	0.7995	0.8901	0.8378

Table 1: Results after training IBM Model 1 with 20 iterations of Expectation Maximization

We notice a drop in precision, while the recall tends to be around the same value. Looking at Figure 1 we infer that our extension tends to work better for short sentences than for longer sentences as compared to the original IBM Model 1.

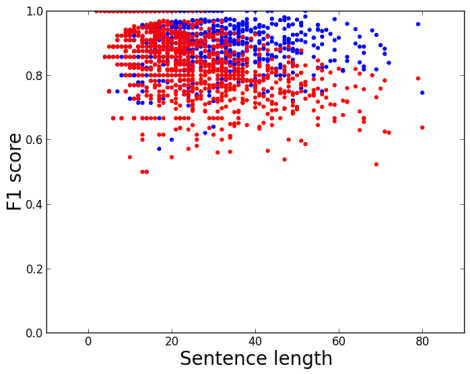

Figure 1: Plot of F1 scores against sentence length. The blue dots denote the original IBM Model 1, while the red dots denote our extension.

5 Conclusion

Despite its structural limitations, such as ignoring positions of words inside sentences or the context surrounding them, IBM Model 1 is still able to achieve good results which make it useful in machine translation tasks.

References

[1] Philipp Koehn 2010. *Statistical Machine Translation*. Cambridge University Press.

[2] Brown, Peter F and Pietra, Vincent J Della and Pietra, Stephen A Della and Mercer, Robert L 1993. *The mathematics of statistical machine translation: Parameter estimation*. Computational linguistics.

[3] Moore, Robert C. 2004 *Improving IBM word-alignment model 1*. Association for Computational Linguistics.

A Translation Table

	Trained:	Viterbi:
NULL		
.	0.81	0.9
het	0.12	0.05
de	0.067	0.05
eigenlijke	0.012	0
onttrokken	0.012	0
aan	3.39e-07	0
significante	3.96e-08	0
meren	3.96e-08	0
territoriale	3.96e-08	0
verbod	3.96e-08	0
the		
de	0.80	0.97
transparantie	0.64	0
het	0.16	0.03
eigenlijke	0.10	0
onttrokken	0.10	0
van	0.04	0
reële	0.01	0
territoriale	0.00	0
significante	0.00	0
meren	0.00	0
in		
in	0.64	0.94
de	0.17	0.02
onderhoud	0.12	0.00
zeewaardigheidsattesten	0.12	0.00
afleveren	0.12	0.00
er	0.08	0.03
van	0.07	0.00
op	0.03	0
significante	0.02	0
meren	0.02	0

Table 2: The top 10 scoring translations according to the training, and corresponding Viterbi probabilities.

	Trained:	Viterbi:
.		
.	0.85	0.99
het	0.15	0.01
eigenlijke	0.01	0
onttrokken	0.01	0
de	0.01	0
aan	2.24e-07	0
is	6.72e-08	0
significante	4.43e-08	0
meren	4.43e-08	0
territoriale	4.43e-08	0
all		
alle	0.55	0.79
duidelijke	0.36	0.05
moeten	0.22	0.13
termijn	0.08	0
een	0.06	0
en	0.06	0.003
begeven	0.05	0
reële	0.05	0
die	0.05	0
leiden	0.04	0
public		
hline en	0.23	0.5
zo	0.17	0.5
met	0.15	0
transparante	0.13	0
komen	0.13	0
onafhankelijke	0.10	0
opinie	0.10	0
concrete	0.08	0
echter	0.08	0
keer	0.08	0

Table 3: Continuation of Table 2. Same caption.