Lecture Notes in Artificial Intelligence, vol. 2499, 54–63, 2002

Efficient integration of maximum entropy models within a maximum likelihood training squeme of statistical machine translation models

Ismael García Varea[1], Franz J. Och[2], Hermann Ney[2], and Francisco Casacuberta[3]

[1] Dpto. de Inf., Univ. of Castilla-La Mancha, 02071 Albacete, Spain
ivarea@info-ab.uclm.es
[2] Lehrstuhl für Inf. VI, RWTH Aachen, Ahornstr., 55 D-52056 Aachen, Germany
[3] Inst. Tecnológico de Inf., Univ. Politécnica de Valencia, 46071 Valencia, Spain

Abstract. Maximum entropy (ME) models has been successfully applied to many natural language problems. In this paper we present how to integrate efficiently ME models within a maximum likelihood trainig scheme of statistical machine translation models. Specifically, we define a set of context-dependent ME lexicon models and we present how to perform an efficient training of these ME models within the conventional expectation-maximization (EM) training of statistical translation models. Experimental results are also presented in order to demonstrate how these ME improve the results obtained with the traditional translation models. The results are presente by means of alignment quality comparing the resulting alignments with a manually annotated reference alignments.

1 Introduction

The ME approach has been applied in natural language processing and machine translation to a variety of tasks. [1] applies this approach to the so-called IBM Candide system to build context-dependent models, to compute automatic sentence splitting and to improve word reordering in translation. Similar techniques are used in [2] for so-called direct translation models instead of those proposed in [3]. [4] use ME models to reduce translation test perplexities and translation errors using a rescoring algorithm, which is applied to n-best translation hypotheses. [5] describes two methods for incorporating information about the relative position of bilingual word pairs into a ME translation model. Other authors have applied this approach to language modeling [6].

In this paper we present how to integrate efficiently ME models within a maximum likelihood trainig scheme of statistical machine translation models. Specifically, we define a set of context-dependent ME lexicon models and we present how to perform an efficient training of these ME models within the conventional EM training of statistical translation models [3]. In each iteration of the training process, the set of ME models is automatically generated by

using the set of possible word-alignments between each pair of sentences. The ME models are trained with the GIS algorithm, then used in the next iteration of the EM training process in order to recompute a new set of parameters of the alignment and lexicon models.

Experimental results are presented for the French-English Canadian Parliament Hansards corpus and the Verbmobil task. The evaluation is performed by comparing the Viterbi alignments obtained after the training of the conventional and the integrated approaches with manually annotated reference alignment.

2 Statistical Machine Translation

The goal of the translation process in statistical machine translation can be formulated as follows: A source language string $\mathbf{f} = f_1^J = f_1 \ldots f_J$ is to be translated into a target language string $\mathbf{e} = e_1^I = e_1 \ldots e_I$. Every target string is considered as a possible translation for the source language string with maximum a-posteriori probability $Pr(\mathbf{e}|\mathbf{f})$. According to Bayes' decision rule, we have to choose the target string that maximizes the product of both the target language model $Pr(\mathbf{e})$ and the string translation model $Pr(\mathbf{f}|\mathbf{e})$. Alignment models to structure the translation model are introduced in [3]. These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. In statistical alignment models, $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$, the alignment $\mathbf{a}$ is introduced as a hidden variable.

The translation probability $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ can be rewritten as follows:

$$\begin{aligned} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) &= \prod_{j=1}^{J} Pr(f_j, a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \\ &= \prod_{j=1}^{J} \left(Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \cdot Pr(f_j | f_1^{j-1}, a_1^j, e_1^I) \right) \ . \end{aligned} \tag{1}$$

3 Conventional EM Training (review)

In this section, we describe the training of the model parameters. Every model has a specific set of free parameters. For example, the parameters θ for Model 4[3] consist of lexicon, alignment and fertility parameters:

$$\theta = \left\{ \{p(f|e)\}, \{p_{=1}(\Delta j)\}, \{p_{>1}(\Delta j)\}, \{p(\phi|e)\}, p_1 \right\} \ . \tag{2}$$

To train the model parameters θ, we pursue a maximum likelihood approach using a parallel training corpus consisting of S sentence pairs $\{(\mathbf{f}_s, \mathbf{e}_s) : s = 1, \ldots, S\}$:

$$\hat{\theta} = \arg\max_{\theta} \prod_{s=1}^{S} \sum_{\mathbf{a}} p_\theta(\mathbf{f}_s, \mathbf{a}|\mathbf{e}_s) \ . \tag{3}$$

We do this by applying the EM algorithm. The different models are trained in succession on the same data, where the final parameter values of a simpler model serve as the starting point for a more complex model.

In the E-step, the lexicon parameter counts for one sentence pair $(\mathbf{e}, \mathbf{f})$ are calculated:

$$c(f|e; \mathbf{e}, \mathbf{f}) = \sum_{\mathbf{e},\mathbf{f}} N(\mathbf{e}, \mathbf{f}) \cdot \sum_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{j} \delta(f, f_j)\delta(e, e_{a_j}) \ . \tag{4}$$

Here, $N(\mathbf{e}, \mathbf{f})$ is the training corpus count of the sentence pair $(\mathbf{f}, \mathbf{e})$.

In the M-step, we want to compute the lexicon parameters $\hat{p}(f|e)$ that maximize the likelihood on the training corpus. This results in the following reestimation [3]:

$$p(f|e) = \frac{\sum_s c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}{\sum_{s,f} c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})} \ . \tag{5}$$

Similarly, the alignment and fertility probabilities can be estimated for all other alignmnent models [3]. When bootstrapping from a simpler model to a more complex model, the simpler model is used to weigh the alignments and the counts are accumulated for the parameters of the more complex model.

4 Maximum Entropy Modeling

4.1 Motivation

Typically, the probability $Pr(f_j|f_1^{j-1}, a_1^j, e_1^I)$ in Equation 1 is approximated by a lexicon model $p(f_j|e_{a_j})$ by dropping the dependencies on f_1^{j-1}, a_1^{j-1}, and e_1^I. Obviously, this simplification is not true for many natural language phenomena. The straightforward approach to include more dependencies in the lexicon model would be to add additional dependencies (e.g. $p(f_j|e_{a_j}, e_{a_{j-1}})$). This approach would yield a significant data sparseness problem. For this reason, we define a set of context-dependent ME lexicon models, which is directly integrated into a conventional EM training of the statistical translation models.

In this case, the role of ME is to build a stochastic model that efficiently takes a larger context into account. In the remainder of the paper, we shall use $p_e(f|x)$ to denote the probability that the ME model (which is associated to e) assigns to f in the context x. Please note that the ME model must be distinguished by the basic lexicon model $p(f|e)$.

4.2 Maximum Entropy Principle

In the ME approach, we describe all properties that we deem to be useful by so-called feature functions $\phi_{e,k}(x, f), k = 1, \ldots, K_e$. For example, let us suppose we want to model the existence or absence of a specific word e'_k in the context of an

English word e, which can be translated by f'_k. We can express this dependence using the following feature function:

$$\phi_{e,k}(x,f) = \begin{cases} 1 \text{ if } f = f'_k \text{ and } e'_k \in x \\ 0 \text{ otherwise} \end{cases} . \tag{6}$$

The ME principle suggests that the optimal parametric form of a model $p_e(f|x)$ taking into account the feature functions $\phi_{e,k}, k = 1, \ldots, K_e$ is given by:

$$p_e(f|x) = \frac{1}{Z_{\Lambda_e}(x)} \exp\left(\sum_{k=1}^{K_e} \lambda_{e,k}\phi_{e,k}(x,f)\right) . \tag{7}$$

Here, $Z_{\Lambda_e}(x)$ is a normalization factor. The resulting model has an exponential form with free parameters $\Lambda_e \equiv \{\lambda_{e,k}, k = 1, \ldots, K_e\}$. The parameter values that maximize the likelihood for a given training corpus can be computed using the so-called GIS algorithm (general iterative scaling) or its improved version IIS [7, 1].

It is important to stress that, in principle, we obtain one ME model for each target language word e. To avoid data sparseness problems for rarely seen words, we use only words that have been seen a certain number of times.

4.3 Contextual Information and Feature Definition

As in [1] we use as a window of 3 words to the left and 3 words to the right of the target word as contextual information. As in [4], in addition to a dependence on the words themselves, we also use, a dependence on the word classes. Thereby, we improve the generalization of the models and include some semantic and syntactic information.

Table 1 summarizes the feature functions that we use for a specific pair of aligned words (e_i, f_j): Category 1 features depend only on the source word f_j and the target word e_i. Categories 2 and 3 describe features that also depend on an additional word e' that appears one position to the left or to the right of e_i, respectively. The features of category 4 and 5 depend on an additional target word e' that appears in any position of the context x. Analogous features are defined using the word class associated to each word instead of the word identity.

To reduce the number of features, we perform a threshold-based feature selection. Every feature that occurs less than T times is not used. The aim of the feature selection is two-fold. Firstly, we obtain smaller models by using less features. Secondly, we hope to avoid overfitting on the training data. In addition, we use ME modeling for target words that are seen at least 150 times.

5 Integrated EM-ME Training

5.1 Training Integration

Using a ME lexicon model for a target word e, we have to train the model parameters $\Lambda_e \equiv \{\lambda_{e,k} : k = 1, \ldots, K_e\}$ instead of the parameters $\{p(f|e)\}$.

Table 1. Meaning of different feature categories where $\square$ represents a specific target word (to be placed in $\bullet$) and $\diamond$ represents a specific source word.

Category	$\phi_{e_i}(x, f_j) = 1$ if and only if ...
1	$f_j = \diamond$
2	$f_j = \diamond$ and $\square \in$ $\begin{array}{\|c\|c\|c\|c\|c\|c\|c\|}\hline & & \bullet & e_i & & & \\ \hline\end{array}$
3	$f_j = \diamond$ and $\square \in$ $\begin{array}{\|c\|c\|c\|c\|c\|c\|c\|}\hline & & & e_i & \bullet & & \\ \hline\end{array}$
4	$f_j = \diamond$ and $\square \in$ $\begin{array}{\|c\|c\|c\|c\|c\|c\|c\|}\hline \bullet & \bullet & \bullet & e_i & & & \\ \hline\end{array}$
5	$f_j = \diamond$ and $\square \in$ $\begin{array}{\|c\|c\|c\|c\|c\|c\|c\|}\hline & & & e_i & \bullet & \bullet & \bullet \\ \hline\end{array}$

We pursue the following approach. In the E-step, we perform a refined count collection for the lexicon parameters:

$$c(f|e, x; \mathbf{e}, \mathbf{f}) = \sum_{\mathbf{e},\mathbf{f}} N(\mathbf{e}, \mathbf{f}) \cdot \sum_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{j} \delta(f, f_j)\delta(e, e_{a_j})\delta(x, x_{j,a_j}) \quad . \quad (8)$$

Here, x_{j,a_j} should denote the ME context that surrounds f_j and e_{a_j}.

In the M-step, we want to compute the lexicon parameters that maximize the likelihood:

$$\hat{\Lambda}_e = \arg\max_{\Lambda_e} \prod_{f,x} c(f|e, x; \mathbf{e}, \mathbf{f}) \cdot \log p(f|e, x) \quad . \quad (9)$$

Hence, the refined lexicon counts $c(f|e, x; \mathbf{e}, \mathbf{f})$ are the weights of the set of training samples (f, e, x) which are used to train the ME models. In Equation 9 $p(f|e, x) \equiv p_e(f|x)$.

The re-estimation of the alignment and fertility probabilities does not change if we use a ME lexicon model.

Thus, we obtain the following steps of each iteration for the EM algorithm:

1. E-step:
   - Collect counts for alignment and fertility parameters.
   - Collect refined lexicon counts.
2. M-step:
   - Re-estimate alignment and fertility parameters.
   - Perform GIS training for lexicon parameters.

5.2 Efficient Training

In a normal iteration of the EM algorithm, in the E-step, a count event collection is performed for the set of considered parameters. Specifically, for the case of the lexicon probabilities the Equations 4 and 8 (for the ME case) are computed summing up the number of times that the word f is translated by e according to the set of possible alignments of each sentence pair of the training corpus. This counts are then used in the M-step to obtain a new refined set of parameters according to the maximum likelihood criterion by using Equations 5 and 9 for the conventional lexicon model and the ME lexicon model respectively.

The problem we are faced in the case of the ME training is that the context x on where the corresponding words e and f appear within the corpus have to be used because the translation probability depends on it. Obviously the E-step has to be performed for every sentence pair in the corpus, and after that in the M-step update the estimation of the parameters for every (e, f) word pair in the input and output vocabularies.

To make this efficiently, a matrix of lexicon probabilities is precomputed for each sentence pair. This matrix contains the probability of every possible connection/translation for each pair of words (e_i, f_j) within a pair of sentences $(\mathbf{e}, \mathbf{f})$, that is, the lexicon probabilities re-estimated in a previous iteration of the training process. In this way we can perfectly distingish between the different context on where each pair of words (e, f) appears for each sentence pair in the corpus. Then we are able to perform exactly the sophisticated count collection for the ME models.

One the E-step is carried out for each sentence pair in the corpus we have all possible ME events (f, e, x) for each word e. Then with these such events we can perform a GIS training for every e word we considered (a priori) relevant to our problem and them obtain the set of Λ_e parameters that define our specific ME model.

In the next iteration of the EM training we will be able to compute the $p_e(f|x)$ by using the ME parameters obtained in the previous iteration. In this case we will also make use of the translation matrix probability which bring us the efficient and easily extraction of the context needed for computing the ME lexicon probability of the specific word pair (e, f).

Another problem with ME modelling is the efficient computation of the normalization factor $Z_{\Lambda_e}(x)$ of Equation 7. The easy identification of the context x also help us to efficiently compute this factor. We only need to sum up the probability of every possible translation of the word e observed in the events (f, e, x) used in the count collection step.

The overload on computation that this approach includes three terms:

1. The identification of the context x for each word translation pair (e, f), which can be computed in a linear computing time due to the use of the translation probability matrices.
2. The additional time due to the ME events generation. This time is despreciable with respect the conventional E-step because only overload on a constant time needed to store each event count to be used a posteriori for the GIS trainig algorithm.
3. The overload included for the GIS training. In this case we will need to perform a GIS training for each word e to be modeled by ME. In the worst case, when all words $e \in \mathcal{V}_e$ (vocabulary of e) are used, the computational time of each iteration of the EM algorithm is increased by the factor $O(GIS * |\mathcal{V}_e|)$.

In the experiments we have carried out the computation time of the GIS algorithm it is in the order of very few second (5 sec. on average). Hence, the computation overload will depend on the number of words e to be modelled by

ME. As we commented at the end of Section 4.3 we develop a ME model for those words that appear (within the training corpus) more than a fixed number of times. This word selection yields on a 10% of words over the vocabulary size. Taking that into account the overall overload will aproximatelly in the order of $O(GIS * |\mathcal{V}_e| * 0.1)$.

A simplification of the approach described above can be obtained in the following way: First, perform a normal training of the EM algorithm. Then, after the final iteration, perform the ME training of the ME lexicon parameters but using only the Viterbi alignment of each sentence pair instead of the set of all possible alignments. Finally, a new EM training is performed where the lexicon parameters are fixed to the ME lexicon models obtained previously. In this case the more informative contextual information is also used but in a decoupled way from the point of view of the EM training. It is important to stress that in this approximation only one ME training is needed, then the overloading computation required from the fully integrated approach is avoided.

6 Experimental Results

We present results on the Verbmobil task and the Hansards task. The Verbmobil task is a speech translation task in the domain of appointment scheduling, travel planning, and hotel reservation. The task is difficult because it consists of spontaneous speech and the syntactic structures of the sentences are less restricted and highly variable. The French-English Hansards task consists of the debates in the Canadian Parliament. This task has a very large vocabulary of more than 100,000 French words.

The corpus statistics are shown in Table 2. The number of running words and the vocabularies are based on full-form words including the punctuation marks. We produced smaller training corpora by randomly choosing 500, 8000 and 34000 sentences from the Verbmobil task and 500, 8000 and 128000 sentences from the Hansards task.

To train the context-dependent statistical alignment models, we extended the publicly available toolkit GIZA++ [8]. The training of the ME models was carried out using the YASMET toolkit [8].

6.1 Evaluation Methodology

We use the same annotation scheme for single-word based alignments and a corresponding evaluation criterion as described in [9]. The annotation scheme explicitly allows for ambiguous alignments. The people performing the annotation are asked to specify two different kinds of alignments: a S (sure) alignment, which is used for alignments that are unambiguous and a P (possible) alignment, which is used for ambiguous alignments. The P label is used particularly to align words within idiomatic expressions, free translations, and missing function words ($S \subseteq P$).

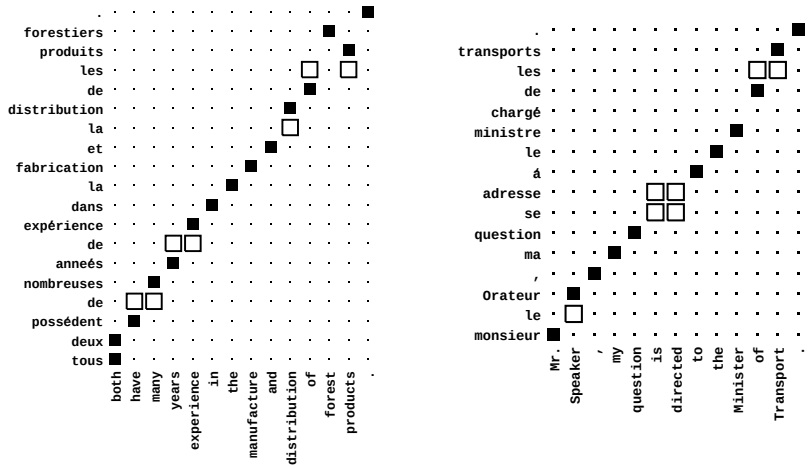

Fig. 1. Two examples of a manual alignment with *S(ure)* (■) and *P(ossible)* (□) connections.

The reference alignment thus obtained may contain many-to-one and one-to-many relationships. Figure 1 shows an example of a manually aligned sentence with S and P labels.

The quality of an alignment $A = \{(j, a_j)|a_j > 0\}$ is then computed by appropriately redefined precision and recall measures and the alignment error rate, which is derived from the well known F-measure:

$$recall = \frac{|A \cap S|}{|S|}, \;\; precision = \frac{|A \cap P|}{|A|}, \;\; AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

Thus, a recall error can only occur if a S(ure) alignment is not found. A precision error can only occur if the alignment found is not even P(ossible).

The set of sentence pairs, for which the manual alignment is produced, is randomly selected from the training corpus. It should be emphasized that all the training is done in a completely unsupervised way, i.e. no manual alignments are used. From this point of view, there is no need to have a separate test corpus.

Table 2. Corpus characteristics.

	Verbmobil		Hansards	
	German	English	French	English
Train Sentences	34446		1470K	
Words	329625	343076	24.33M	22.16M
Vocabulary	5936	3505	100269	78332

Table 3. AER [%] on Hansards (left) and Verbmobil (right) tasks.

		Size of training corpus							
Train. scheme	Model		0.5K	8K	128K		0.5K	8K	34K
1^5	1	Handsards	48.0	35.1	29.2	Verbmobil	27.7	19.2	17.6
	1+ME		47.7	32.7	22.5		24.6	16.6	13.7
$1^5 2^5$	2		46.0	29.2	21.9		26.8	15.7	13.5
	2+ME		44.7	28.0	19.0		25.3	14.1	10.8
$1^5 2^5 3^3$	3		43.2	27.3	20.8		25.6	13.7	10.8
	3+ME		42.5	26.4	17.2		24.1	11.6	8.8
$1^5 2^5 3^3 4^3$	4		41.8	24.9	17.4		23.6	10.0	7.7
	4+ME		41.3	24.3	14.1		22.8	9.3	7.0
$1^5 2^5 3^3 4^3 5^3$	5		41.5	24.8	16.2		22.6	9.9	7.2
	5+ME		41.2	24.3	14.3		22.3	9.6	6.8

6.2 Alignment Quality Results

Table 3 shows the alignment quality for different training sample sizes of the Hansards and Verbmobil tasks. This table shows the baseline AER for different training schemes and the corresponding values when the integration of the ME is done. The training scheme is defined in accordance with the number of iterations performed for each model (4^3 means 3 iterations of Model 4).

The recall and precision results for the Hansards task with and without ME training are shown in Figure 2.

We observe that the alignment error rate improves when using the context-dependent lexicon models. For the Verbmobil task, the improvements were smaller than for the Hansards task, which might be due to the fact that the baseline alignment quality was already very good. It can be seen that larger improvements were obtained for the simpler models.

As expected, the ME training takes a more important role when larger sizes of the corpus are used. For the smallest corpora, the number of training events for the ME models is very low, so it is not possible to disambiguate some translations/alignments for different contexts. For larger sizes of the corpora, greater improvements are obtained. Therefore, we expect to obtain better improvements when using even larger corpora.

7 Conclusions

In this paper, we present an efficient and straightforward integration of ME context-dependent models within a maximum likelihood training of statistical translation models.

We evaluate the quality of the alignments obtained with this new training scheme comparing the results with the baseline results. As can be seen in Section 6, we obtain better alignment quality using the context-dependent lexicon model.

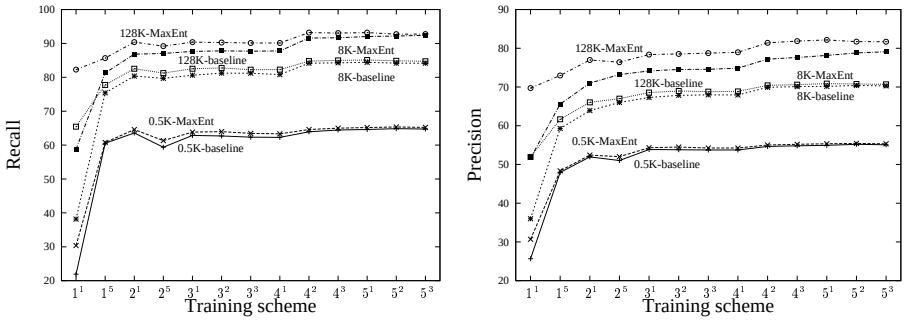

Fig. 2. Recall and Precision [%] results for Hansards task for different corpus sizes, for every iteration of the translation squeme.

In the future, we plan to include more features in the ME model, such us dependencies with other source and target words, POS tags and syntactic constituents. We also plan to design ME alignment and fertility models. This will allow for an easy integration of more dependencies, such as second-order alignment models without running into the problem of an unmanageable number of alignment parameters. We have just started to perform experiments for a very distant pair of languages like Chinese-English with very promising results.

References

[1] Berger, A.L., Pietra, S.A.D., Pietra, V.J.D.: A maximum entropy approach to natural language processing. Computational Linguistics **22** (1996) 39–72

[2] Papineni, K., Roukos, S., Ward, R.: Maximum likelihood and discriminative training of direct translation models. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing. (1998) 189–192

[3] Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1993) 263–311

[4] García-Varea, I., Och, F.J., Ney, H., Casacuberta, F.: Refined lexicon models for statistical machine translation usign a maximum entropy approach. In: Proc. of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France (2001) 204–211

[5] Foster, G.: Incorporating position information into a maximum entropy/minimum divergence translation model. In: Proc. of CoNNL-2000 and LLL-2000, Lisbon, Portugal (2000) 37–52

[6] Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modeling. Computer, Speech and Language **10** (1996) 187–228

[7] Pietra, S.D., Pietra, V.D., Lafferty, J.: Inducing features in random fields. IEEE Trans. on Pattern Analysis and Machine Inteligence **19** (1997) 380–393

[8] Och, F.J., Ney, H.: Giza++: Training of statistical translation models (2000) `http://www-i6.Informatik.RWTH-Aachen.DE/~och/software/GIZA++.html`.

[9] Och, F.J., Ney, H.: A comparison of alignment models for statistical machine translation. In: COLING '00: The 18th Int. Conf. on Computational Linguistics, Saarbrücken, Germany (2000) 1086–1090