Searching for alignments in SMT. A novel approach based on an Estimation of Distribution Algorithm [*]

Luis Rodríguez, Ismael García-Varea, José A. Gámez
Departamento de Sistemas Informáticos
Universidad de Castilla-La Mancha
luisr@dsi.uclm.es, ivarea@dsi.uclm.es, jgamez@dsi.uclm.es

Abstract

In statistical machine translation, an alignment defines a mapping between the words in the source and in the target sentence. Alignments are used, on the one hand, to train the statistical models and, on the other, during the decoding process to link the words in the source sentence to the words in the partial hypotheses generated. In both cases, the quality of the alignments is crucial for the success of the translation process. In this paper, we propose an algorithm based on an Estimation of Distribution Algorithm for computing alignments between two sentences in a parallel corpus. This algorithm has been tested on different tasks involving different pair of languages. In the different experiments presented here for the two word-alignment shared tasks proposed in the HLT-NAACL 2003 and in the ACL 2005, the EDA-based algorithm outperforms the best participant systems.

1 Introduction

Nowadays, statistical approach to machine translation constitutes one of the most promising approaches in this field. The rationale behind this approximation is to learn a statistical model from a parallel corpus. A parallel corpus can be defined as a set of sentence pairs, each pair containing a sentence in a source language and a translation of this sentence in a target language. Word alignments are necessary to link the words in the source and in the target sentence. Statistical models for machine translation heavily depend on the concept of alignment, specifically, the well known IBM word based models (Brown et al., 1993). As a result of this, different task on aligments in statistical machine translation have been proposed in the last few years (HLT-NAACL 2003 (Mihalcea and Pedersen, 2003) and ACL 2005 (Joel Martin, 2005)).

In this paper, we propose a novel approach to deal with alignments. Specifically, we address the problem of searching for the best word alignment between a source and a target sentence. As there is no efficient exact method to compute the optimal alignment (known as *Viterbi alignment*) in most of the cases (specifically in the IBM models 3,4 and 5), in this work we propose the use of a recently appeared meta-heuristic family of algorithms, *Estimation of Distribution Algorithms* (EDAs). Clearly, by using a heuristic-based method we cannot guarantee the achievement of the optimal alignment. Nonetheless, we expect that the global search carried out by our algorithm will produce high quality results in most cases, since previous experiments with this technique (Larrañaga and Lozano, 2001) in different optimization task have demonstrated. In addition to this, the results presented in section 5 support the approximation presented here.

This paper is structured as follows. Firstly, Statistical word alignments are described in section 2. Estimation of Distribution Algorithms (EDAs) are

[*] This work has been supported by the Spanish Projects JCCM (PBI-05-022) and HERMES 05/06 (Vic. Inv. UCLM)

Proceedings of the Workshop on Statistical Machine Translation, pages 47–54,
New York City, June 2006. ©2006 Association for Computational Linguistics

introduced in section 3. An implementation of the search for alignments using an EDA is described in section 4. In section 5, we discuss the experimental issues and show the different results obtained. Finally, some conclussions and future work are discussed in section 6.

2 Word Alignments In Statistical Machine translation

In statistical machine translation, a word alignment between two sentences (a source sentence $\mathbf{f}$ and a target sentence $\mathbf{e}$) defines a mapping between the words $f_1 ... f_J$ in the source sentence and the words $e_1 .. e_I$ in the target sentence. The search for the optimal alignment between the source sentence $\mathbf{f}$ and the target sentence $\mathbf{e}$ can be stated as:

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a} \in A} Pr(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \operatorname*{argmax}_{\mathbf{a} \in A} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \quad (1)$$

being A the set of all the possible alignments between $\mathbf{f}$ and $\mathbf{e}$.

The transformation made in Eq. (1) allows us to address the alignment problem by using the statitistical approach to machine translation described as follows. This approach can be stated as: 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 regarded 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 (Brown et al., 1993). These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \to i = a_j$ 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 usually introduced as a hidden variable. Nevertheless, in the problem described in this article, the source and the target sentences are given, and we are focusing on the optimization of the aligment $\mathbf{a}$.

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

$$\begin{aligned} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) &= \prod_{j=1}^{J} Pr(f_j, a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \\ &= \prod_{j=1}^{J} Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \\ &\quad \cdot Pr(f_j | f_1^{j-1}, a_1^{j}, e_1^I) \qquad (2) \end{aligned}$$

The probability $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ can be estimated by using the word-based IBM statistical alignment models (Brown et al., 1993). These models, however, constrain the set of possible alignments so that each word in the source sentence can be aligned at most to one word in the target sentence. Of course, "real" alignments, in most of the cases, do not follow this limitation. Hence, the alignments obtained from the IBM models have to be extended in some way to achieve more realistic alignments. This is usually performed by computing the alignments in both directions (i.e, first from $\mathbf{f}$ to $\mathbf{e}$ and then from $\mathbf{e}$ to $\mathbf{f}$) and then combining them in a suitable way (this process is known as symmetrization).

3 Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) (Larrañaga and Lozano, 2001) are metaheuristics which has gained interest during the last five years due to their high performance when solving combinatorial optimization problems. EDAs, as well as genetics algorithms (Michalewicz, 1996), are population-based evolutionary algorithms but, instead of using genetic operators are based on the estimation/learning and posterior sampling of a probability distribution, which relates the variables or genes forming and individual or chromosome. In this way the dependence/independence relations between these variables can be explicitly modelled in the EDAs framework. The operation mode of a canonical EDA is shown in Figure 1.

As we can see, the algorithm maintains a population of m individuals during the search. An individual is a candidate or potential solution to the problem being optimized, e.g., in the problem considered here an individual would be a possible alignment. Usually, in combinatorial optimization problems an individual is represented as a vector of integers $\mathbf{a} = \langle a_1, \ldots, a_J \rangle$, where each position a_j can

```
1. $D_0 \leftarrow$ Generate the initial population ($m$ individuals)
2. Evaluate the population $D_0$
3. $k = 1$
4. Repeat
    (a) $D_{tra} \leftarrow$ Select $s \leq m$ individuals from $D_{k-1}$
    (b) Estimate/learn a new model $\mathcal{M}$ from $D_{tra}$
    (c) $D_{aux} \leftarrow$ Sample $m$ individuals from $\mathcal{M}$
    (d) Evaluate $D_{aux}$
    (e) $D_k \leftarrow$ Select $m$ individuals from $D_{k-1} \cup D_{aux}$
    (f) $k = k + 1$
   Until stop condition
```

Figure 1: A canonical EDA

take a set of finite values $\Omega_{a_j} = \{0, \ldots, I\}$. The first step in an evolutionary algorithm is to generate the initial population D_0. Although D_0 is usually generated randomly (to ensure diversity), prior knowledge can be of utility in this step.

Once we have a population our next step is to evaluate it, that is, we have to measure the goodness or fitness of each individual with respect to the problem we are solving. Thus, we use a fitness function $f(\mathbf{a}) = Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ (see Eq. (3)) to score individuals. Evolutionary algorithms in general and EDAs in particular seek to improve the quality of the individuals in the population during the search. In genetic algorithms the main idea is to build a new population from the current one by copying some individuals and constructing new ones from those contained in the current population. Of course, as we aim to improve the quality of the population with respect to fitness, the best/fittest individuals have more chance to be copied or selected for recombination.

In EDAs, the transition between populations is quite different. The basic idea is to summarize the properties of the individuals in the population by learning a probability distribution that describes them as much as possible. Since the quality of the population should be improved in each step, only the s fittest individuals are selected to be included in the dataset used to learn the probability distribution $Pr(\mathbf{a}_1, \ldots, \mathbf{a}_J)$, in this way we try to discover the common regularities among good individuals. The next step is to obtain a set of new individuals by sampling the learnt distribution. These individuals are scored by using the fitness function and added to the ones forming the current population. Finally, the new population is formed by selecting n individuals from the $2n$ contained in the current one. A common practice is to use some kind of fitness-based elitism during this selection, in order to guarantee that the best(s) individual(s) is/are retained.

The main problem in the previous description is related to the estimation/learning of the probability distribution, since estimating the joint distribution is intractable in most cases. In the practice, what is learnt is a probabilistic model that consists in a factorization of the joint distribution. Different levels of complexity can be considered in that factorization, from univariate distributions to n-variate ones or Bayesian networks (see (Larrañaga and Lozano, 2001, Chapter 3) for a review). In this paper, as this is the first approximation to the alignment problem with EDAs and, because of some questions that will be discussed later, we use the simplest EDA model: the *Univariate Marginal Distribution Algorithm* or UMDA (Muhlenbein, 1997). In UMDA it is assumed that all the variables are marginally independent, thus, the n-dimensional probability distribution, $Pr(a_1, \ldots, a_J)$, is factorized as the product of J marginal/unidimensional distributions: $\prod_{j=1}^{J} Pr(a_j)$. Among the advantages of UMDA we can cite the following: no structural learning is needed; parameter learning is fast; small dataset can be used because only marginal probabilities have to be estimated; and, the sampling process is easy because each variable is independently sampled.

4 Design of an EDA to search for alignments

In this section, an EDA algorithm to align a source and a target sentences is described.

4.1 Representation

One of the most important issues in the definition of a search algorithm is to properly represent the space of solutions to the problem. In the problem considered here, we are searching for an "optimal" alignment between a source sentence $\mathbf{f}$ and a target sentence $\mathbf{e}$. Therefore, the space of solutions can be stated as the set of possible alignments between both sentences. Owing to the constraints imposed by the IBM models (a word in $\mathbf{f}$ can be aligned at most to one word in $\mathbf{e}$), the most natural way to represent a solution to this problem consists in storing each possible alignment in a vector $\mathbf{a} = a_1 ... a_J$, being J the length of $\mathbf{f}$. Each position of this vector can take the value of "0" to represent a NULL alignment (that is, a word in the source sentence that is aligned to no words in the target sentence) or an index representing any position in the target sentence. An example of alignment is shown in Figure 4.1.

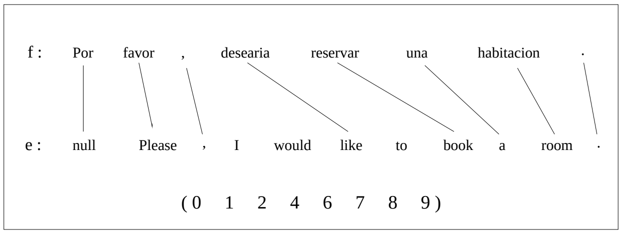

Figure 2: Example of alignment and its representation as a vector

4.2 Evaluation function

During the search process, each individual (search hypothesis) is scored using the fitness function described as follows. Let $\mathbf{a} = a_1 \cdots a_J$ be the alignment represented by an individual. This alignment $\mathbf{a}$ is evaluated by computing the probability $p(\mathbf{f}, \mathbf{a}|\mathbf{e})$. This probability is computed by using the IBM model 4 as:

$$
\begin{aligned}
p(\mathbf{f}, \mathbf{a}|\mathbf{e}) = & \sum_{(\tau,\pi) \in \langle \mathbf{f}, \mathbf{a} \rangle} p(\tau, \pi | \mathbf{e}) \\
& \prod_{i=1}^{I} n(\phi_i | e_i) \times \prod_{i=1}^{I} \prod_{k=1}^{\phi_i} t(\tau_{ik}|e_i) \times \\
& \prod_{i=1, \phi_i > 0}^{I} d_{=1}(\pi_{i1} - c_{\rho_i} | \mathcal{E}_c(e_{\rho_i}), \mathcal{F}_c(\tau_{i1})) \times \\
& \prod_{i=1}^{I} \prod_{k=2}^{\phi_i} d_{>1}(\pi_{ik} - \pi_{i(k-1)} | \mathcal{F}_c(\tau_{ik})) \times \\
& \binom{J - \phi_0}{\phi_0} p_0^{J-2\phi_0} p_1^{\phi_0} \times \prod_{k=1}^{\phi_0} t(\tau_{0k}|e_0) \qquad (3)
\end{aligned}
$$

where the factors separated by × symbols denote fertility, translation, head permutation, non-head permutation, null-fertility, and null-translation probabilities[1].

This model was trained using the GIZA++ toolkit (Och and Ney, 2003) on the material available for the different alignment tasks described in section 5.1

4.3 Search

In this section, some specific details about the search are given. As was mentioned in section 3, the algorithm starts by generating an initial set of hypotheses (initial population). In this case, a set of randomly generated alignments between the source and the target sentences are generated. Afterwards, all the individuals in this population (a fragment of a real population is shown in figure 3) are scored using the function defined in Eq.(4.2). At this point, the actual search starts by applying the scheme shown in section 3, thereby leading to a gradual improvement in the hypotheses handled by the algorithm in each step of the search.

This process finishes when some finalization criterium (or criteria) is reached. In our implementation, the algorithm finishes when it passes a certain number of generations without improving the quality of the hypotheses (individuals). Afterwards, the best individual in the current population is returned as the final solution.

Regarding the EDA model, as commented before, our approach rely on the UMDA model due mainly to the size of the search space defined by the task. The algorithm has to deal with individuals of length J, where each position can take $(I + 1)$ possible values. Thus, in the case of UMDA, the number of free parameters to be learnt for each position is I (e.g., in the English-French task $avg(J) = 15$ and $avg(I) = 17.3$). If more complex models were considered, the size of the probability tables would have grown exponentially. As an example, in a bivariate model, each variable (position) is conditioned on another variable and thus the probability tables $P(.|.)$ to be learnt have $I(I + 1)$ free parameters. In order to properly estimate the probabilty distributions, the size of the populations has to be increased considerably. As a result, the computational resources

[1] The symbols in this formula are: J (the length of $\mathbf{e}$), I (the length of $\mathbf{f}$), e_i (the i-th word in e_1^I), e_0 (the NULL word), ϕ_i (the fertility of e_i), τ_{ik} (the k-th word produced by e_i in $\mathbf{a}$), π_{ik} (the position of τ_{ik} in $\mathbf{f}$), ρ_i (the position of the first fertile word to the left of e_i in $\mathbf{a}$), c_{ρ_i} (the ceiling of the average of all $\pi_{\rho_i k}$ for ρ_i, or 0 if ρ_i is undefined).

```
1 1 5 3 2 0 6 0    (-60.7500)
1 6 5 2 3 0 0 5    (-89.7449)
1 2 2 6 4 0 5 0    (-90.2221)
1 2 3 5 0 3 6 2    (-99.2313)
0 6 0 2 4 6 3 5    (-99.7786)
2 0 0 2 2 0 3 4    (-100.587)
1 0 1 6 3 6 0 5    (-101.335)
```

Figure 3: Part of one population generated during the search for the alignments between the English sentence *and then he tells us the correct result !* and the Romanian sentence *si ne spune noua rezultatul corect !*. These sentences are part of the HLT-NAACL 2005 shared task. Some individuals and their scores (fitness) are shown.

required by the algorithm rise dramatically.

Finally, as was described in section 3, some parameters have to be fixed in the design of an EDA. On the one hand, the size of each population must be defined. In this case, this size is proportional to the length of the sentences to be aligned. Specifically, the size of the population adopted is equal to the length of source sentence **f** multiplied by a factor of ten.

On the other hand, as we mentioned in section 3 the probability distribution over the individuals is not estimated from the whole population. In the present task about 20% of the best individuals in each population are used for this purpose.

As mentioned above, the fitness function used in the algorithm just allows for unidirectional alignments. Therefore, the search was conducted in both directions (i.e, from $\mathbf{f}$ to $\mathbf{e}$ and from $\mathbf{e}$ to $\mathbf{f}$) combining the final results to achieve bidirectional alignments. To this end, diffferent approaches (symmetrization methods) were tested. The results shown in section 5.2 were obtained by applying the *refined method* proposed in (Och and Ney, 2000).

5 Experimental Results

Different experiments have been carried out in order to assess the correctness of the search algorithm. Next, the experimental metodology employed and the results obtained are described.

5.1 Corpora and evaluation

Three different corpora and four different test sets have been used. All of them are taken from the two shared tasks in word alignments developed in HLT/NAACL 2003 (Mihalcea and Pedersen, 2003) and ACL 2005 (Joel Martin, 2005). These two tasks involved four different pair of languages, English-French, Romanian-English, English-Inuktitut and English-Hindi. English-French and Romanian-English pairs have been considered in these experiments (owing to the lack of timeto properly preprocess the Hindi and the Inuktitut). Next, a brief description of the corpora used is given.

Regarding the Romanian-English task, the test data used to evaluate the alignments consisted in 248 sentences for the 2003 evaluation task and 200 for the 2005 evaluation task. In addition to this, a training corpus, consisting of about 1 million Romanian words and about the same number of English word has been used. The IBM word-based alignment models were training on the whole corpus (training + test). On the other hand, a subset of the Canadian Hansards corpus has been used in the English-French task. The test corpus consists of 447 English-French sentences. The training corpus contains about 20 million English words, and about the same number of French words. In Table 1, the features of the different corpora used are shown.

To evaluate the quality of the final alignments obtained, different measures have been taken into account: *Precision*, *Recall*, *F-measure*, and *Alignment Error Rate*. Given an alignment A and a reference alignment G (both A and G can be split into two subsets A_S,A_P and G_S, G_P, respectively representing *Sure* and *Probable* alignments) *Precision* (P_T), *Recall* (R_T), *F-measure* (F_T) and *Alignment Error Rate* (AER) are computed as (where T is the alignment type, and can be set to either S or P):

$$
\begin{aligned}
P_T &= \frac{|A_T \bigcap G_T|}{|A_T|} \\
R_T &= \frac{|A_T \bigcap G_T|}{|G_T|} \\
F_T &= \frac{|2P_T R_T|}{|P_T + R_T|} \\
AER &= \frac{1 - |A_S \bigcap G_S| + |A_P \bigcap G_P|}{|A_P| + |G_S|}
\end{aligned}
$$

Table 1: Features of the corpora used in the different alignment task

	En-Fr	**Ro-En 03**	**Ro-En 05**
Training size	1M	97K	97K
Vocabulary	68K / 86K	48K / 27K	48K / 27K
Running words	20M / 23M	1.9M / 2M	1.9M / 2M
Test size	447	248	200

It is important to emphasize that EDAs are non-deterministics algorithms. Because of this, the results presented in section 5.2 are actually the mean of the results obtained in ten different executions of the search algorithm.

5.2 Results

In Tables 2, 3 and 4 the results obtained from the different tasks are presented. The results achieved by the technique proposed in this paper are compared with the best results presented in the shared tasks described in (Mihalcea and Pedersen, 2003) (Joel Martin, 2005). The results obtained by the GIZA++ hill-climbing algorithm are also presented. In these tables, the mean and the variance of the results obtained in ten executions of the search algorithm are shown. According to the small variances observed in the results we can conclude that the non-deterministic nature of this approach it is not statistically significant.

According to these results, the proposed EDA-based search is very competitive with respect to the best result presented in the two shared task.

In addition to these results, additional experiments were carried out in to evaluate the actual behavior of the search algorithm. These experiments were focused on measuring the quality of the algorithm, distinguishing between the errors produced by the search process itself and the errors produced by the model that leads the search (i.e, the errors introduced by the fitness function). To this end, the next approach was adopted. Firstly, the (bidirectional) reference alignments used in the computation of the Alignment Error Rate were split into two sets of unidirectional alignments. Owing to the fact that there is no exact method to perform this decomposition, we employed the method described in the following way. For each reference alignment, all the possible decompositions into unidirectional alignments were perfomed, scoring each of them with the evaluation function $F(\mathbf{a}) = p(\mathbf{f}, \mathbf{a}|\mathbf{e})$ defined in section (3), and being selected the best one, $\mathbf{a}_{ref}$. Afterwards, this alignment was compared with the solution provided by the EDA, $\mathbf{a}_{eda}$. This comparison was made for each sentence in the test set, being measuried the AER for both alignments as well as the value of the fitness function. At this point, we can say that a model-error is produced if $F(\mathbf{a}_{eda}) > F(\mathbf{a}_{ref})$. In addition, we can say that a search-error is produced if $F(\mathbf{a}_{eda}) < F(\mathbf{a}_{ref})$. In table 5, a summary for both kinds of errors for the English-Romanian 2005 task is shown. In this table we can also see that these results correlate with the AER figures.

These experiments show that most of the errors were not due to the search process itself but to another different factors. From this, we can conclude that, on the one hand, the model used to lead the search should be improved and, on the other, different techniques for symmetrization should be explored.

6 Conclusions and Future Work

In this paper, a new approach, based on the use of an Estimation of Distribution Algorithm has been presented. The results obtained with this technique are very promising even with the simple scheme here considered.

According to the results presented in the previous section, the non-deterministic nature of the algorithm has not a real influence in the performance of this approach. Therefore, the main theoretical drawback of evolutionary algorithms have been proven not to be an important issue for the task we have addressed here.

Finally, we are now focusing on the influence of these improved alignments in the statistical models for machine translation and on the degree of accu-

Table 2: Alignment quality (%) for the English-French task with NULL alignments

System	P_s	R_s	F_s	P_p	R_p	F_p	AER
EDA	**73.82**	82.76	**78.04**	**83.91**	29.50	43.36	**13.61** ±0.03
GIZA++	73.61	82.56	77.92	79.94	32.96	46.67	15.89
Ralign.EF1	72.54	80.61	76.36	77.56	**36.79**	**49.91**	18.50
XRCE.Nolem.EF.3	55.43	**93.81**	69.68	72.01	36.00	48.00	21.27

Table 3: Alignment quality (%) for the Romanian-English 2003 task with NULL aligments

System	P_s	R_s	F_s	P_p	R_p	F_p	AER
EDA	94.22	49.67	65.05	76.66	60.97	**67.92**	**32.08** ±0.05
GIZA++	**95.20**	48.54	64.30	**79.89**	57.82	67.09	32.91
XRCE.Trilex.RE.3	80.97	53.64	64.53	63.64	**61.58**	62.59	37.41
XRCE.Nolem-56k.RE.2	82.65	**54.12**	**65.41**	61.59	61.50	61.54	38.46

Table 4: Alignment quality (%) for the Romanian-English 2005 task

System	P_s	R_s	F_s	P_p	R_p	F_p	AER
EDA	95.37	54.90	69.68	80.61	67.83	**73.67**	**26.33** ±0.044
GIZA++	**95.68**	53.29	68.45	81.46	65.83	72.81	27.19
ISI.Run5.vocab.grow	87.90	63.08	**73.45**	87.90	63.08	73.45	26.55
ISI.Run4.simple.intersect	94.29	57.42	71.38	**94.29**	57.42	71,38	28.62
ISI.Run2.simple.union	70.46	**71.31**	70.88	70.46	**71.31**	70.88	29.12

Table 5: Comparison between reference aligments (decomposed into two unidirectional alignments) and the alignments provided by the EDA. Search errors and model errors for EDA and GIZA++ algorithms are presented. In addition, the AER for the unidirectional EDA and reference alignments is also shown. These result are obtained on the Romanian-English 05 task

	Romanian-English	**English-Romanian**
EDA search errors (%)	35 (17.5 %)	18 (9 %)
EDA model errors (%)	165 (82.5 %)	182 (91 %)
GIZA++ search errors (%)	87 (43 %)	81 (40 %)
GIZA++ model errors (%)	113 (57 %)	119 (60 %)
AER-EDA	29.67 %	30.66 %
AER-reference	12.77 %	11.03 %

racy that could be achieved by means of these aligments. In addition to this, the integration of the aligment algorithm into the training process of the statistical translation models is currently being performed.

References

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comp. Linguistics*, 19(2):263–311.

Ted Pedersen Joel Martin, Rada Mihalcea. 2005. Word alignment for languages with scarce resources. In Rada Mihalcea and Ted Pedersen, editors, *Proceedings of the ACL Workshop on Building and Exploiting Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10, Michigan, USA, June 31. Association for Computational Linguistics.

P. Larrañaga and J.A. Lozano. 2001. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers.

Z. Michalewicz. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In Rada Mihalcea and Ted Pedersen, editors, *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–10, Edmonton, Alberta, Canada, May 31. Association for Computational Linguistics.

Heinz Muhlenbein. 1997. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346.

Franz J. Och and Hermann Ney. 2000. Improved statistical alignment models. In *ACL00*, pages 440–447, Hongkong, China, October.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.