Applied Artificial Intelligence
An International Journal

Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713191765

EDA: AN EVOLUTIONARY DECODING ALGORITHM FOR STATISTICAL MACHINE TRANSLATION
Eridan Otto a, María Cristina Riff a
a Departamento de Informática, Universidad Técnica Federico Santa María, Valparaíso, Chile

Online Publication Date: 01 August 2007
To cite this Article: Otto, Eridan and Riff, María Cristina (2007) ‘EDA: AN EVOLUTIONARY DECODING ALGORITHM FOR STATISTICAL MACHINE TRANSLATION’, Applied Artificial Intelligence, 21:7, 605 - 621
To link to this article: DOI: 10.1080/08839510701492546
URL: http://dx.doi.org/10.1080/08839510701492546

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article maybe used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
EDA: AN EVOLUTIONARY DECODING ALGORITHM FOR STATISTICAL MACHINE TRANSLATION

Eridan Otto and María Cristina Riff  
Department de Informática, Universidad Técnica Federico Santa María, Valparaíso, Chile

In a statistical machine translation system (SMTS), decoding is the process of finding the most likely translation based on a statistical model, according to previously learned parameters. The success of an SMTS is strongly dependent on the quality of its decoder. Most of the SMTSs published in current literature use approaches based on traditional optimization methods and heuristics. On the other hand, over the last few years there has been a rapid increase in the use of meta-heuristics. These kinds of techniques have shown to be able to solve difficult search problems in an efficient way for a wide number of applications.

This paper proposes a new approach based on evolutionary hybrid algorithms to translate sentences in a specific technical context. The algorithm has been enhanced by adaptive parameter control. The tests are carried out in the context of Spanish and then translated to English.

The experimental results validate the superior performance of our method in contrast to a statistical greedy decoder. We also compare our new approach to the existing public domain general translators.

Machine translation (MT) is the process of automatic translation from one natural language to another using a computer program. Human language consists of morphology, syntax, and semantics. It is often argued that the problem of MT requires the problem of natural language understanding to be solved first. However, a number of empirical methods of translation work surprisingly well (Arnold et al. 1995). Between these methods we find statistical-based methods, which try to generate translations based on bilingual text corpora. Statistical machine translation (SMT) was first introduced by Brown et. al. (1993) in the 90s. In order to design an SMT that can translate a source sentence $s$ (for example Spanish) into a target sentence $e$ (for example English), the following components are required.

This work was carried out in the context of the Fondecyt Project 1060377.
Address correspondence to M. C. Riff, Departamento de Informática, Universidad Técnica Federico Santa María, Av. España 1680, Casilla 110-V, Valparaíso, Chile. E-mail: mcriff@inf.utfsm.cl
A language model (LM) that assigns a probability $P(e)$ to each English string

- A translation model (TM) that assigns a probability $P(s|e)$ to each pair of English and Spanish string
- A decoder that uses for input a new sentence $s$ and tries to generate a translated sentence $e$ as output, which maximizes the translation probability $P(e|s)$, or according to the Bayes rule equivalently maximizes $P(e) \cdot P(s|e)$.

Efficient algorithms to estimate the probabilities for a language model, like n-grams models (Chen and Goodman 1996), exist. Translation models are usually based on word replacement models, which were developed by IBM in the early 1990s (Brown et al. 1990; Brown et al. 1993; Brown et al. 1995; Berger et al. 1996). These models are referred to as (IBM) models 1–5. In this paper, we focus our attention on the decoder. A good decoding or search algorithm is critical for the success of any SMT system (German, et al. 2004). Knight (1999) has shown that a decoding problem is an NP-complete. Thus, because of decoding problem complexities, optimal decoders, i.e., decoders that guarantee finding optimal solutions, are not used in practical SMT implementation. However, some approximated decoders have been proposed in the literature, as stack or $\Delta^*$ algorithms (Brown et al. 1995; Wang and Waibel 1997, Och and Ney 2006), dynamic programming-based algorithms (Tillman and Ney 2000, Garcia-Varea and Casacuberta 2001; Watanabe and Sumita 2002), and greedy heuristic-based algorithms (Germann et al. 2004; Germann 2003).

From our knowledge of the metaheuristics community, only decoders based on greedy methods have been implemented. In this paper, we introduce an evolutionary decoding algorithm, in order to improve the efficiency of the translation task in the SMT framework. The translation is performed from Spanish to English sentences, in the context of the computer science technical area. A comparison of the results obtained between applying a public greedy decoder and our algorithm is presented. Both decoders use the same learning model. Finally, the translations obtained using our evolutionary algorithm as a decoder are compared to other general domain public translators.

**STATISTICAL MACHINE TRANSLATION**

With a few exceptions (Vogel et al. 1996) most SMT systems are based on the noisy channel framework (see Figure 1). The approach is inspired on the success of statistical techniques applied to speech recognition. In the machine translation framework, the sentences $e$ is written in a source language, for example, English, it is then supposed to be transformed by
a noisy probabilistic channel that generates the equivalent target sentences $s$, which in our case in Spanish. For the rest of this paper we refer to the source language as the language into which the SMT translates.

The two keys notions involved are those of the language model and the translation model. The language model provides us with probabilities for strings of words or sentences $P(e)$, which are estimated using a monolingual corpus. The source vocabulary $e$ is defined by the set of all different words in the source corpus, analogously for the target vocabulary $S$.

Broadly speaking, the probabilities for each sentence from source corpus is independently calculated. The translation model provides us with conditional probabilities. In order to estimate the conditional probability $P(s|e)$ that occurs in a target sentence $s$, the target text which translates a text containing the source sentence $e$, requires a large bilingual aligned corpus.

Usually the parameters of both the language and the translation models are estimated using traditional maximum likelihood and expectation maximization techniques (Dempster et al. 1977). Translation is the problem of finding the $e$ that is most probable given $s$.

Our work is based on the IBM model 4 translation model (Ueffing et al. 2002). Before we begin to describe this model, it is useful to introduce some further notions. In a word aligned sentence-pair, it is indicated which target words correspond to each source word, as shown in Figure 2.
For that, the following variables are required.

- $l$ the number of words of $e$, $m$ the number of words of $s$
- $e_i$ the $i$th word of $e$, $s_j$ the $j$th word of $s$

It is important to note that a source word can be aligned with more than one word in the target sentence. The fertility of a source word is determined by the number of words corresponding to it in the target string. In theory, an alignment can correspond to any set of connections. However, IBM’s models are restricted to alignments where each target word at most corresponds to one source word. It is possible to represent the alignment $a$ as a vector $(a_1, a_2, \ldots, a_m)$, where the value of $a_k$ indicates the word position in the source sentence that corresponds to the $k$th word in the target sentence. In the case of a target word that is not connected to any source word, the $a_k$ value is equal to zero. In Figure 2 we illustrate this using a NULL symbol.

**IBM Model 4**

Our work is based on the IBM model 4, which is shown in Figure 3. This model uses the following sub-models to calculate the conditional probabilities $P(a, s|e)$.

- **Lexical model** $t(s_j|e_i)$: Word-for-word translation model, representing the probability of a word $s_j$ corresponding to the word $e_i$, which means $e_i$ is aligned with $s_j$.
- **Fertility model** $n(\phi_i|e_i)$: Representing the probability of a source word $e_i$ is aligned to $\phi_i$ words in the target sentence $s$. When a source word is not aligned to a target word, $\phi_i$ is equal to zero.
- **Distortion model** $d$: This model captures the probability that the position of a word in the source language changes in the target language. For this the model uses a cluster technique to define word classes $A(e_i)$ for the source language and word classes $B(s_j)$ for the target language.

<table>
<thead>
<tr>
<th>Translation Model</th>
<th>Lexical Model</th>
<th>Fertility Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\prod t(s_j</td>
<td>e_{a_j})$</td>
<td>$\prod n(\phi_i</td>
</tr>
</tbody>
</table>

| Distortion Model | $\prod d_1(j - c_{p_1}|A(e_{p_1}), B(s_j))$ |
|------------------|---------------------------------------------|
| Head             | Non-Head $\prod d_{>1}(j - j'|B(s_j))$ |

| NULL Translation Model | $(\frac{m - \phi_0}{\phi_0}) p_1^{\phi_0}(1 - p_1)^{m - 2\phi_0}$ |

**FIGURE 3** Translation model (IBM model 4).
The words are clustering using some similarity criteria (Och 1999). The distortion model is also broken down into two sets of parameters:

- **Distortion probability for head words**: A head word is the first $s_j$ word aligned with $e_i$, with fertility $\phi_i$ not equal to zero. It is denoted by the following expression: $d_1(j - c_{p_j}|A(e_{p_j}), B(s_j))$, where $j$ is the head word position, $p_j$ is the position of the first word fertile to the left of $e_i$, and $c_{p_j}$ is the representative position of the word in $s$ of its alignment with $e_{p_j}$. In case of word $e_{p_j}$, fertility is greater than one; its representative position is calculated as the upper boundary of the average of all word positions aligned with it.

- **Distortion probability for non-head words**: $d_{\neq 1}(j - j' | B(s_j))$: If the word $e_i$ has a fertility greater than one, $j$ represents the position of a non-head word and $j'$ is the position of the first word to the left of $s_j$, which is aligned with $e_i$.

- **NULL translation model $p_1$**: This allows for the calculation of the probability the number of words of $s$ aligned to NULL value in the target sentence. Finally, the $P(a, s|e)$ is calculated by multiplying all the sub-models probabilities described. See Brown et al. (1993) and Germann et al. (2004) for a detailed discussion of this translation model and a description of its parameters.

**Decoder Problem**

In this paper we focus our attention on the decoding problem. Knight (1999) has shown that for an arbitrary word-reordering, the decoding problem is NP-complete. Roughly speaking, a decoder takes as input a new sentence $s$ and tries to generate a translated sentence $e$ as output, which maximizes the translation probability $P(e|s)$, where

$$P(e|s) = P(e)P(s|e)$$

and

$$P(s|e) = \sum_a P(a, s|e)$$

is the addition of $P(a, s|e)$ over all possible alignments $a$. In practice it is infeasible to consider all $m_{l+1}$ alignments. Thus, an approximately value of $P(s|e) \sim P(a, s|e)$ is used to find the most probable $(e, a)$ that maximizes $P(e)p(a, s|e)$.

**EVOLUTIONARY DECODING ALGORITHM**

Evolutionary algorithms (EAs) are probabilistic search techniques. Given an optimization problem, they try to find an optimal solution (Goldberg 1989). EAs start by initializing a set of possible solutions called
individuals. It needs a fitness function to evaluate these solutions in order to distinguish between better and worse individuals. An EA is an iterative process that tries to improve the average fitness of a set of individuals or population by applying a transformation procedure to a set of selected individuals to construct a new population. After some criterion is met, the algorithm returns the best individuals of the population. In this section we present the components of an evolutionary decoding algorithm (EDA), specifically designed to solve the decoding problem.

**Individual Representation**

A chromosome is composed of two related structures: an alignment structure and a translation structure. The alignment structure is set according to the definition given previously, which is a vector of a fixed length of integers \( a_j \). The translation structure is a string of variable lengths of token \( e_j \) and it represents a translated sentence. Figure 4 shows the representation of the example of the Figure 2.

This representation contains all information that the algorithm requires to do a fast evaluation and to use specialized genetic operators.

**Initial Population**

To generate the initial population we develop a greedy randomized construction heuristic. This method attempts to assure diversity of the initial population. Due to its random component the algorithm begins with a population of individuals that represents different points of the space of translations. On the other hand, the greedy component of the algorithm looks for good initial solutions selecting at each instantiation a word from a list of the more appropriate English words for the Spanish ones. The algorithm is presented in Figure 5.

**Fitness Function**

According to the problem definition, the value to be maximized is \( P(e)P(s|e) \). In order to have a better and significant discrimination between the fitness values, we define the following evaluation function:

\[
\hat{e} = \arg \min_{e} (-\log(P(e)) - \log(P(s|e)))
\]  

(3)
The \textit{log} function is monotonic, thus the lowest fitness value should correspond to the best translation. It works according to the training parameters of both models. Furthermore, it simplifies the partial evaluations of both models and sub-models.

Selection

We apply the well-known roulette wheel method (Holland 1975), where the selection probability of an individual is proportional to its fitness function value. EDA uses elitism, thus the best individual is passed directly to the next generation.

Specialized Recombination Operators

We designed three different recombination operators. The goal is to exchange translation information between the parents to create a new better individual. All of these operators create two offspring, but select the best of them to continue to the next generation. Each operator has an independent probability to be applied. The idea is to use dynamic adaptive parameter control mechanisms in order for the algorithm itself to able to change its parameters values depending on the evolution.

- **One point alignment crossover**: This operator makes a crossover on the alignment part of the representation. The translation structure is changed according to the new alignment obtained in each offspring. The procedure is shown in Figure 6.
- **Lexical exchange crossover**: This is a fast operator that focuses on the exchange lexical components of the chromosome. Both children inherit an alignment structure from their parents. In order to construct the child
translation structure, synonymous words from both parents are interchanged according to its alignment.

- **Greedy lexical crossover**: This algorithm tries to find a good alignment. Each child has the same alignment structure of each parent. In order to construct each child translation structure, the best translated word for \( s_j \) from both parents is selected using the lexical model \( t(s_j|e_i) \). This word is located in each child’s translation structure in the position determined by its alignment.

The Figure 7 shows an example of applying each operator for the same sentence to be translated.

**Specialized Asexual Operators**

We have two classes of asexual operators: a random class of operators designed to help the algorithm to escape from a local optima and a local search class of operator, which is designed to exploit the search space given the problem complexity.

**Random Operators**

We propose two exploration operators, which help the algorithm to escape from a local optima

- **Mutation word**: This operator acts in the translation structure selecting a word \( s_k \). The current word \( e_i \) that translates \( s_k \) is replaced by a randomly selected synonym word from \( RCL_k \).

- **Simple swap**: This operator randomly selects the position of two words on the translation structure and swaps their words. It modifies the alignment structure according to the new words positions.

**Local Search-Based Operators**

Because it is a complex problem (Hart and Belew 1996), we also include two hybrids operators that perform a local search procedure.
Language model local search: It is a local search operator that works on the language model. It is a hill-climbing procedure whose goal is to improve the language model probability that is calculated using trigrams partitions. The operator analyzes each sequence of three consecutive words of the translation structure. It uses a partial evaluation of the six permutations in order to select the best ordering trigram between them. Finally, when all trigrams have been analyzed and probably changed, the algorithm makes a global evaluation to accept or reject the new individual.
**Procedure** Translation Model Local Search (*Chromosome*)

Begin

For each word $e_2$ with zero fertility in $E$

- Tries to insert $e_2$ in the position of the translation structure which gives the best improvement of the evaluation of the language model

For each word $e_i$ with zero fertility in the translation structure

Delete $e_i$ when this action improves the evaluation function

For $i_1$=1 to $l$

For $i_2 = 1$ to $l$

- If $i_1 \neq i_2$

- If $\phi_{i_1} > 1$ and $n(\phi_{i_1} | e_{i_1}) > 0$

  - link all $s_j$ words aligned with $e_{i_2}$ to $e_{i_1}$

  - delete $e_{i_2}$ from the translation structure

End

**FIGURE 8** Structure of translation model local search.

- **Translation model local search**: This is a best improvement-based operator, which uses the features of IBM model 4. It works with the fertility concept. The algorithm is shown in Figure 8. It is an exhaustive procedure which in the beginning tries to insert zero fertility words from the vocabulary in the translation structure. In the second step it deletes zero fertility words included in the translation structure, which increases the evaluation function value. The next step is focused on fertility. The idea is to increase word fertility of the translation structure in order to reduce the number of words in the translation. This augmentation of fertility is accepted only if the evaluation function improves.

**Parameter Control Mechanisms in EDA**

The algorithm manages the most critical parameters as recombination and mutation operator’s probabilities with an adaptive parameter control strategy. The goal is to find the best combination of the parameters changing their values during the execution, according to the state of the search (Riff and Bonnaire 2002). In the beginning all the operator’s probabilities are equal. The algorithm constructs a ranking based on the accumulated statistical information during $g$ generations. It classifies the operators according to their successfulness at finding good offspring. It gives a reward to the operator that produces the better offspring increasing its probability. Therefore, the probabilities of the worse operators are reduced. It is represented by the following equation:

$$P_{i,t+1} = (1 - \alpha) \cdot R_{i,t} + \alpha \cdot P_{i,t}$$  \hspace{1cm} (4)$$

where $P_{i,t}$ is the probability of the operator $i$ in the generation $t$, $R_{i,t}$ is the reward, and $\alpha$ is a momentum parameter used to smooth the change of the probabilities.
EXPERIMENTAL RESULTS

For the translation model we use a bilingual parallel corpus and for the language model we use an English monolingual corpus. The decoder performance is highly dependent on the data quality, the translation model, the language model, and also the algorithms used for training both models. We used articles from the bilingual ACM crossroads magazine (ACM year) to construct the corpus. This magazine is published online in English and Spanish.

Corpus

The construction of a bilingual parallel corpus, that is, a set of sentences in both languages where each sentence is aligned to its corresponding translated sentence, is a very time-consuming task. To accomplish this goal we have completed the following steps.

1. We have selected the Web pages corresponding to the papers of the journal that are available in both languages. We have obtained 38 articles which sizes were 1360 Kb in Spanish and 1240 Kb in English.
2. We have filtered the text to obtain the paragraphs alignment. We took advantage of the HTML tags to find the corresponding sentences.
3. We have designed a program to generate aligned sentences from the aligned paragraphs. This is a crucial task, because introducing mistakes at this step will make the decoder produce results of poor quality. To be sure to make a correct identification of the aligned sentences, we applied some heuristics proposed in Garcia-Varea and Casacuberta (2001), including a metric of the proportion of the length between English and Spanish sentences, considering the number of characters. We also use the interpretation of the punctuation characters as delimiters of the sentences, especially the dot.
4. In order to produce a more precise corpus, we discarded all sentences that had a length greater than 50 words from the sets of aligned sentences in both languages.

The result of this procedure is shown in Table 1. We obtained 4812 bilingual English–Spanish sentences in the context of computer science.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Training and Test Conditions for the Computational Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary</td>
<td>8681</td>
</tr>
<tr>
<td>Training: Sentences</td>
<td>4812</td>
</tr>
<tr>
<td>Words</td>
<td>92933</td>
</tr>
</tbody>
</table>
Training

We have defined five types of tests. All of them to translate 80 sentences from Spanish to English. The 80 sentences have been divided into five groups, according to the number of words: 8, 10, 12, 16, and 18. The difference among the five types of tests is the criteria used for the selection of the sentences from the original corpus to be tested and for the delete action, that is, if the selected sentence is deleted or not from the corpus. Obviously, it will affect the training step. The results for the training corpus and the training sets for each test are shown in Table 2.

The five types of tests are the following.

- **T1:** We have randomly selected and deleted 80 sentences to translate from the original corpus. The key idea of this test is to prove the ability of the decoder to translate unknown sentences. In this test, we only consider words that appear in the training corpus.

- **T2:** Similar to T1, the difference is that the selected sentences are not deleted from the training corpus. The idea of this test is to evaluate the decoder when it translates known sentences. This case is designed to be compared with T1.

- **T3, T4, T5:** In these tests 80 sentences are randomly selected, as in T1. But now, the words either appear or do not appear in the corpus vocabulary. The key idea is to evaluate the performance of the decoder related to the modifications of both the models and the tested sentences.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Vocabulary</th>
<th>Sentences</th>
<th>Words</th>
<th>Vocabulary</th>
<th>Sentences</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>8681</td>
<td>4732</td>
<td>91808</td>
<td>6709</td>
<td>4732</td>
<td>83599</td>
</tr>
<tr>
<td>T2</td>
<td>8681</td>
<td>4812</td>
<td>92933</td>
<td>6721</td>
<td>4812</td>
<td>84650</td>
</tr>
<tr>
<td>T3</td>
<td>8648</td>
<td>4732</td>
<td>91945</td>
<td>6703</td>
<td>4732</td>
<td>83724</td>
</tr>
<tr>
<td>T4</td>
<td>8636</td>
<td>4732</td>
<td>91951</td>
<td>6690</td>
<td>4732</td>
<td>83705</td>
</tr>
<tr>
<td>T5</td>
<td>8639</td>
<td>4732</td>
<td>91947</td>
<td>6693</td>
<td>4732</td>
<td>83733</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish</td>
<td>English</td>
</tr>
<tr>
<td>T1</td>
<td>511</td>
</tr>
<tr>
<td>T2</td>
<td>511</td>
</tr>
<tr>
<td>T3</td>
<td>511</td>
</tr>
<tr>
<td>T4</td>
<td>511</td>
</tr>
<tr>
<td>T5</td>
<td>511</td>
</tr>
</tbody>
</table>
We used the CMU-Cambridge Statistical Language Modeling Toolkit v2 (Clarkson and Rosenfeld 1999) for the training language model. The GIZA++ toolkit permitted us to estimate the parameters of the translation model (Och and Ney 2000a). EDA works with the following tuned parameters: population size \((\text{popsize} = 50)\), maximum number of generations \((\text{maxgens} = 50)\), and \(cl = 15\). The more critical parameters use a dynamic adaptive parameter control mechanism described next with \(\alpha = 0.4\), \(g = 5\), and rewards matrix \(0.5\), \(0.35\), \(0.15\). The hardware platform for the experiments was a PC Pentium III, 870 Mhz, with 256 Mb RAM under Linux operating system. The algorithm has been implemented in C.

**Performance Measures**

There are two widely used metrics (Tillman et al. 1997) to measure quality in MT: word error rate (WER) and position-independent error rate (PER). WER corresponds to the number of transformations (insert, replace, delete) to be done to the translation solution generated by EDA, in order to obtain the reference translation. PER is similar to WER, but it only takes into account the number of replace actions to do.

**Tests**

The algorithm is evaluated using three tests classes: The first one is a translation example from phrases taken from the abstract of this article; see Table 3.

**TABLE 3** Examples of Spanish to English Translations

<table>
<thead>
<tr>
<th>Input</th>
<th>EDA</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a decodificación es el proceso de encontrar 1a traducción más probable.</td>
<td>decoding is the process find the most likely translation.</td>
<td>decoding is the process of finding the most likely translation.</td>
</tr>
<tr>
<td>EDA: 1a decodificación is the process find the most likely translation.</td>
<td>EDA: the success either these system is highly dependent the quality its decodificador.</td>
<td>EDA: the success of any such system is highly dependent on the quality of its decoder.</td>
</tr>
<tr>
<td>Ref: 1a decodificación is the process find the most likely translation.</td>
<td>Ref: the success of any such system is highly dependent on the quality of its decoder.</td>
<td>Ref: the success of any such system is highly dependent on the quality of its decoder.</td>
</tr>
<tr>
<td>el éxito de cualquiera de estos sistemas es altamente dependiente de la calidad de su decodificador.</td>
<td>EDA: this article shows a new approach based on evolutionary algorithms for translate sentences in a tech specific context.</td>
<td>EDA: this article shows a new approach based on evolutionary algorithms to translate statements in a specific technical context.</td>
</tr>
<tr>
<td>EDA: the success either these system is highly dependent the quality its decodificador.</td>
<td>Ref: this article shows a new approach based on evolutionary algorithms for translate sentences in a tech specific context.</td>
<td>Ref: this article shows a new approach based on evolutionary algorithms to translate statements in a specific technical context.</td>
</tr>
<tr>
<td>Ref: the success of any such system is highly dependent on the quality of its decoder.</td>
<td>Input: este artículo muestra un nuevo enfoque basado en algoritmos evolutivos para traducir sentencias en un contexto técnico específico.</td>
<td>Input: este artículo muestra un nuevo enfoque basado en algoritmos evolutivos para traducir sentencias en un contexto técnico específico.</td>
</tr>
<tr>
<td>EDA: this article shows a new approach based on evolutionary algorithms for translate statements in a tech specific context.</td>
<td>EDA: the evolutionary algorithm translates with quality top with respect to transducers general purpose.</td>
<td>EDA: the evolutionary algorithm translates with higher quality respect to a general purpose translators.</td>
</tr>
<tr>
<td>Ref: this article shows a new approach based on evolutionary algorithms to translate statements in a specific technical context.</td>
<td>Ref: sentence taken of the abstract.</td>
<td>Ref: sentence taken of the abstract.</td>
</tr>
</tbody>
</table>
The second one is a comparison between EDA and the greedy decoder *isi rewrite decoder v.0.7*[^3], which is based on the work of Germman et al. (2004). Both decoders used the same language model, the same translation model, and they were trained with the same corpus. Finally, the translations obtained by using our evolutionary algorithm as a decoder are compared to other general domain public translators. The results are documented in Tables 4 and 5.

EDA was able to translate new sentences with different lengths quite well. We can conclude that in the computer science context by using EDA we can obtain an effective algorithm to translate from Spanish to English.

The Table 4 shows EDA outperforms in 60% of the tests of the greedy decoder. We remarked that EDA obtains better alignments of the sentence than the greedy decoder and the translation generated by it is closest to the reference sentence. The results of EDA are more remarkable in WER than PER, because PER takes into account the words, but not the full sentence alignment. The specialized operators on the evolutionary framework enable the algorithm to do a search focused on both the words and their alignment.

Finally, we compared our results with two public domain translators: Babelfish[^4] and SDL International.[^5] Because EDA has a contextual nature, i.e., it’s specifically trained for computer science context, the results

<table>
<thead>
<tr>
<th>Type of algorithm</th>
<th>WER%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence length</td>
<td>8</td>
</tr>
<tr>
<td>T1 Statistical</td>
<td></td>
</tr>
<tr>
<td>EDA</td>
<td>18</td>
</tr>
<tr>
<td>Greedy</td>
<td>22</td>
</tr>
<tr>
<td>General</td>
<td>43</td>
</tr>
<tr>
<td>Purpose</td>
<td>40</td>
</tr>
<tr>
<td>T2 Statistical</td>
<td></td>
</tr>
<tr>
<td>EDA</td>
<td>11</td>
</tr>
<tr>
<td>Greedy</td>
<td>20</td>
</tr>
<tr>
<td>General</td>
<td>43</td>
</tr>
<tr>
<td>Purpose</td>
<td>40</td>
</tr>
<tr>
<td>T3 Statistical</td>
<td></td>
</tr>
<tr>
<td>EDA</td>
<td>21</td>
</tr>
<tr>
<td>Greedy</td>
<td>22</td>
</tr>
<tr>
<td>General</td>
<td>29</td>
</tr>
<tr>
<td>Purpose</td>
<td>33</td>
</tr>
<tr>
<td>T4 Statistical</td>
<td></td>
</tr>
<tr>
<td>EDA</td>
<td>43</td>
</tr>
<tr>
<td>Greedy</td>
<td>43</td>
</tr>
<tr>
<td>General</td>
<td>55</td>
</tr>
<tr>
<td>Purpose</td>
<td>58</td>
</tr>
<tr>
<td>T5 Statistical</td>
<td></td>
</tr>
<tr>
<td>EDA</td>
<td>19</td>
</tr>
<tr>
<td>Greedy</td>
<td>30</td>
</tr>
<tr>
<td>General</td>
<td>43</td>
</tr>
<tr>
<td>Purpose</td>
<td>40</td>
</tr>
</tbody>
</table>

obtained by our SMT are better than the translations generated by these public domain general translators.

CONCLUSIONS

Our research allowed us to conclude that using an evolutionary approach to translate with the statistical machine translation framework, it is feasible and comparable in quality with other techniques using the same framework and other kinds of translators. There are a variety of other statistical translation models and all of them need a decoder to perform the translation. The results suggest that our technique is a good option to implement a decoder adapting EDA to the different features of a specific statistical model. We are moving towards designing recombination operators which use the features of both the translation and the language models.

REFERENCES


NOTES

1. The available collection has trimestral numbers from 1998 to 2002; however, only some of them were in both languages.