Division of Spanish Words into Morphemes with a Genetic Algorithm*

Alexander Gelbukh[1], Grigori Sidorov[1],
Diego Lara-Reyes[1] and Liliana Chanona-Hernandez[2]

[1] Natural Language and Text Processing Laboratory,
Center for Research in Computer Science, National Polytechnic Institute,
Av. Juan Dios Batiz, s/n, Zacatenco, 07738, Mexico City, Mexico
www.Gelbukh.com, www.cic.ipn.mx/~sidorov
[2] ESIME Zacatenco, National Polytechnic Institute,
Zacatenco, 07738, Mexico City, Mexico

Abstract. We discuss an unsupervised technique for determining morpheme structure of words in an inflective language, with Spanish as a case study. For this, we use a global optimization (implemented with a genetic algorithm), while most of the previous works are based on heuristics calculated using conditional probabilities of word parts. Thus, we deal with complete space of solutions and do not reduce it with the risk to eliminate some correct solutions beforehand. Also, we are working at the derivative level as contrasted with the more traditional grammatical level interested only in flexions. The algorithm works as follows. The input data is a wordlist built on the base of a large dictionary or corpus in the given language and the output data is the same wordlist with each word divided into morphemes. First, we build a redundant list of all strings that might possibly be prefixes, suffixes, and stems of the words in the wordlist. Then, we detect possible paradigms in this set and filter out all items from the lists of possible prefixes and suffixes (though not stems) that do not participate in such paradigms. Finally, a subset of those lists of possible prefixes, stems, and suffixes is chosen using the genetic algorithm. The fitness function is based on the ideas of minimum length description, i.e. we choose the minimum number of elements that are necessary for covering all the words. The obtained subset is used for dividing the words from the wordlist. Algorithm parameters are presented. Preliminary evaluation of the experimental results for a dictionary of Spanish is given.

1 Introduction

We present an application of a global optimization technique (implemented as a genetic algorithm) to the task of division of words into morphemes using Spanish language as a case study, i.e., our main goal is investigating the application of the unsupervised technique for determining morpheme structure of words.

Word division into morphemes is useful for automatic description of morphological structures of languages without existing morphological models and/or morphological

* Work done under partial support of Mexican Government (CONACYT, SNI) and National Polytechnic Institute, Mexico (SIP, COFAA, PIFI).

E. Kapetanios, V. Sugumaran, M. Spiliopoulou (Eds.): NLDB 2008, LNCS 5039, pp. 19–26, 2008.

dictionaries. It is important for modern information retrieval technologies, when we are dealing with unknown languages or languages without complete description [5], or, possibly, for some specific terminological areas, say, medicine.

The prevalent approach is presented in [3] and it is implemented in *Linguistica* system. Variations of this method are described in [4], [7], and [1]. The main idea of this approach is to use heuristics for reduction of the algorithm search space. There are two principal heuristics. The first heuristics is related to the construction of the initial set of potential morphemes, which is based on their conditional probabilities. This procedure is iterative. For each word, one division with maximum weight is selected from all possible divisions. For calculation of these weights, the conditional probabilities are used under certain empirical assumptions that are not theoretically justified, for example, the assumption of Boltzmann distribution of probabilities is assumed. The procedure is repeated until the maximum weights are achieved. Then the minimum description length (MDL) technique is applied as a second heuristics to the set of possible signatures (potential paradigms) for their improvement and debugging.

There were two competitions of automatic division into morphemes. In 2005 [9], the abovementioned methods were applied to division of words in various languages (English, Finnish, Turkish); though Spanish was not considered. In 2007 [8], the task was changed to finding the aspects of morpheme meaning.

The idea of detection of repetitive sets of non-stem morphemes (potential paradigms) is vital for all methods. Its' result is also called *signature*, we still prefer more linguistic term *paradigm*; in our case, it is *derivational paradigm*. These terms refer to the fact that stems can be concatenated with sets of morphemes, and these sets are repeated across the vocabulary, for example, *high, highly, highness* and *bright, brightly, brightness* share the set {∅, *-ly*, *-ness*}. It is really surprising that usage of these sets greatly reduces the search space for all types of algorithms.

The idea of our approach is avoid using heuristics and apply a search of the global optimum in the space of all possible solutions. In our case, we implement it as a genetic algorithm. Thus, we deal with complete space of solutions and do not reduce it with the risk to eliminate some correct solutions beforehand. On the other hand, we are working at the derivative level, as contrasted with the more traditional grammatical level, where the interest is centered on flexions only. Spanish language has rather simple morphological structure, so no more than three possible word parts are considered.

There were attempts to apply genetic algorithm to this problem [2], [6]. Still, the results were not very promising because none of these methods took into account possible repetition of sets of morphemes (paradigms). In this paper, we modify the method adding this possibility.

The rest of the paper has the following structure. First, we describe the algorithm, then its' parameters and preliminary experimental results are presented, finally, conclusions are drawn and future work is discussed.

2 Algorithm

In this section we present description of the algorithm and discuss its' parameters used in the experiments.

2.1 Algorithm Description

The general scheme of the algorithm is presented in Fig. 1.

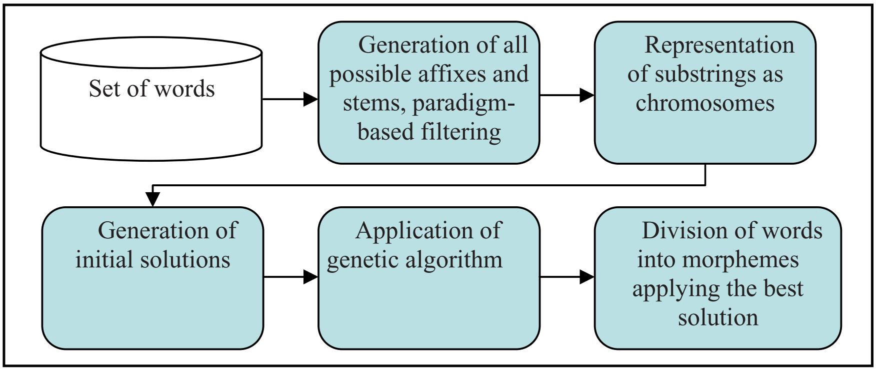
Set of words
Generation of all possible affixes and stems, paradigm-based filtering
Representation of substrings as chromosomes
Generation of initial solutions
Application of genetic algorithm
Division of words into morphemes applying the best solution

Fig. 1. Algorithm scheme

Detailed description of algorithm is as follows:

1. For each word, we detect all possible prefixes including the empty string ∅, starting from the first letter and finishing at the penultimate letter. The strings are added to the list of possible prefixes without repetitions along with their frequencies.
2. We prepare the list of possible stems, starting from the first letter to the last letter. The stem list contains unique elements with their corresponding frequencies and does not include the empty string ∅.
3. The list of possible suffixes is prepares taking the substrings starting from the last letter till the second letter from the beginning. This list contains the empty string ∅ and also does not permit repetitions. It also includes frequencies of suffixes.
4. All possible paradigms are detected for stems, i.e., the sets of morphemes that are repeated several times for various stems. For example, some paradigms are {NULL, *-ism, -iz, -idad*}, {NULL, *-ant, -acion*}. We filter out all non-stem substrings (potential morphemes) that are not part of some stem paradigm that contains more than one element (these sets can be considered also as the paradigms with only one element, but they are not really a paradigm, but a unique combination). Also, we filter out morphemes that belong only to paradigms with low frequency. In our case, we used values of frequencies of two and three.
5. Chromosomes (individuals) are formed according to the following rules:
   a. They are binary in a sense that their genes are represented as “0” and “1”, thus, the chromosome is a sequence of genes.
   b. Each gene (binary position in chromosome) corresponds to a string from one of the lists,

   c. Thus, the length of the chromosome is the sum of the number of elements in the three lists.
   d. In the chromosome, value “1” means that the corresponding string of the corresponding list is part of the solution, while value “0” means that it does not participate in word formation for this solution.
6. Initial population consisting of several chromosomes is generated in a random way.
7. Genetic algorithm is applied with various parameters (see below). Note that if mutation or crossover generates a chromosome that is not “valid”, i.e. it contains stems that do not participate in division of any valid word then the chromosome is “improved” by adding in a random way some affixes that correct this problem.

   This is the necessary step since we cannot rely only on selection, because the evaluation is performed for the whole chromosome (solution), and, thus, some incorrect divisions can stay until the final stage because they can be compensated by other high valued genes.
8. Fitness function of each chromosome is calculated in the following way:
   a. The number of the genes used in the solution (non-zero genes) is as small as possible;
   b. The combination of substrings used in the solution from all three lists covers major number of words from the initial vocabulary. All possible combinations of prefixes, suffixes and stems presented in the given chromosome are verified.
9. When the algorithm finishes, the best solution is used to divide words from the initial vocabulary into morphemes.

Traditional parameters of the genetic algorithm that should be mentioned are:

1. Selection procedures. Genetic operator of selection is executed using the tournament scheme, namely, for two randomly chosen individuals their fitness is compared and the best individual wins. This guarantees that the individuals are competing and that the better ones survive.
2. Crossover. This genetic operator is implemented using as its’ parameter a number of blocks on the basis of which the chromosomes exchange their genetic information for creation of new individuals. The places for crossover are chosen randomly.
3. Replacement of individuals by the created ones. We used the elitist replacement scheme, when the best individuals are always conserved in the population.
4. Mutation. This genetic operator is important because it allows for creation of the new possibilities in the space of solutions. It changes values of the randomly chosen genes with certain probability.

2.2 Algorithm Parameters

We conducted several experiments with genetic algorithm and compared their results. We found out that the best results are obtained using three different sets of parameters consequently in three passes of the algorithm. Application of several passes is a common way of usage of genetic algorithms, when at each pass a specific purpose should be achieved.

We experimented within the following ranges of parameters, see Table 1.

Table 1. Ranges of parameters for the genetic algorithm

	Minimal values	Maximal values
Population size	50	5,000
Replacement	20%	100%
Mutation	Starting from 20%, reducing according to number of generations	Starting from 90%, reducing according to number of generations
Crossover	1	20
Generations	50	10,000

The following sets of parameters for the genetic algorithm give the best results applyed in the consecutive manner (three passes) for population of 200 individuals; see Tables 2, 3 and 4.

Table 2. Parameters for first pass

Replacement	60%
Mutation	30%, reducing according to number of generations
Crossover	5
Generations	10,000

Table 3. Parameters for second pass

Replacement	80%
Mutation	80%, reducing according to number of generations
Crossover	20
Generations	7,000

Table 4. Parameters for third pass

Replacement	40%
Mutation	20%, reducing according to number of generations
Crossover	4
Generations	6,000

The purpose of each pass is different. At the first pass, the population is prepared and regularized. At the second pass, the population is shaken at the maximum grade. Finally, at the third pass, the populations should stabilize, for example, mutation rate is significantly reduced.

3 Experimental Results

We used as input data a Spanish dictionary that contained more than 20,000 head words. We ignored auxiliary words and adverbs, thus, working only with nouns, verbs and adjectives. Since we are interested in derivative morphology, we cut off their flexions, for example*, trabajar → trabaj- (to work)*, *rojo → roj- (red)*, etc. Also, we treated the stressed vowels indistinctly to the non-stressed vowels because they have purely orthographic function in Spanish. The final input list contained 16,849 unique non-flexional words.

For the moment, for being able to perform a comparison with *Linguistica* system (J. Goldsmith, [3]), we made the experiments for suffixes only, though the algorithms permits simultaneous treatment of suffixes and prefixes.

The following results were obtained while preparing the lists of initial strings for the algorithm. We found 7,747 derivational paradigms, from which 6,472 contained more than one element. Thus, the paradigms that contained exactly one element were ignored. It is worth mentioning that of these 6,472: 1,852 paradigms contained zero string. Also we used a threshold for paradigm repetition, namely, we ignored paradigms that repeated less than three times, i.e., they exist for three words or less. There were 5,535 paradigms with frequency equal to 1; 404 with frequency equal to 2; and 171 with frequency equal to 3, etc. Finally, only 372 paradigms left for processing in the algorithm. They were used for filtering, i.e., only suffixes and stems that participated in them are used for representation of chromosomes. Finally, the algorithm worked with 17,085 stems and 136 suffixes. It is a really important reduction taking into account that initial number of stems was more than 44,000 and the same number of suffixes was more than 15,000.

We compared our results with the output produced by *Linguistica* system, giving it the same input. Unfortunately, we do not have the golden standard of our data for automatic evaluation. Manual evaluation of results shows that our system produced comparable division of words during the evaluation procedure described below.

The version of *Linguistica* that we have, accepts only 5,000 words as input. So we gave to both systems the same input: first 5,000 words from our dictionary. This number should be large enough because both systems use unsupervised learning. Then we evaluate manually the obtained divisions of the first hundred words. *Linguistica* has 87% of precision, while our system produces 84%. The systems have errors in different words. For example, *Linguistica* did not find the suffix *–mient(o)* that was rather frequent in the list and was detected by our system. Obviously, these values of precision correspond only to a preliminary estimation. These values may seem too high because the baseline of the existing systems is around 60%, but let us remember that we are dealing with derivative morphology and we are working with the dictionary, not with the corpus. The exact evaluation remains as a task for future work.

4 Conclusions and Future Work

We described an application of an optimization technique, namely, genetic algorithm, to the problem of division of word into morphemes. Our primary concern was derivative morphology and we made our experiments for Spanish language as a case study.

In the algorithm, the chromosome is constructed from a set of all possible substrings for stems and affixes filtered in a special way. The genes in the chromosomes are binary, when "1" means the presence of the element (stem or affix), and "0" corresponds to its' absence. We used as the filter the presence of a substring in paradigm with frequency greater than three. Also we ignored the paradigms that consisted of only one element.

Traditional genetic algorithm operators are applied to processing of chromosomes (=individuals) in populations.

Our results are comparable with more traditional techniques based on heuristics with calculations of conditional probabilities, but our method uses all space of solutions and do not filter out some possibly correct solutions beforehand.

For the moment, we are filtering out all non-stem substrings (potential morphemes) that are not part of some stem paradigm. In future, we would like to try different treatment of these paradigms, namely, including the paradigms in evaluation of fitness function or including them somehow directly in chromosomes. Also, we plan trying simultaneous treatment of suffixes and prefixes as parts of paradigms.

The future work is also related with performing the exact evaluation, though we are not aware of the golden standard for Spanish derivational morphology. We plan to develop this standard and use it for automatic evaluation.

References

[1] Baroni, M., Matiasek, J., Trost, H.: Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In: ACL Workshop on Morphological and Phonological Learning (2002)

[2] Gelbukh, A., Alexandrov, M., Han, S.: Detecting Inflection Patterns in Natural Language by Minimization of Morphological Model. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 432–438. Springer, Heidelberg (2004)

[3] Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics 27(2), 153–198 (2001)

[4] Creutz, M.: Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Sapporo, Japan, pp. 280–287 (2003)

[5] Haahr, P., Baker, S.: Making search better in Catalonia, Estonia, and everywhere else. Google blog (2007), http://googleblog.blogspot.com/2008/03/making-search-better-in-catalonia.htm

[6] Kazakov, D.: Unsupervised learning of naıve morphology with genetic algorithms. In: Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks. Prague, Czech Republic, pp. 105–112 (1997)

[7] Rehman, K., Hussain, I.: Unsupervised Morphemes Segmentation. In: Pascal Morphochallenge, 5p. (2005)

[8] Pascal Morphochallenge (2007), http://www.cis.hut.fi/morphochallenge2007/

[9] Pascal Morphochallenge (2005), http://www.cis.hut.fi/morphochallenge2005/