Unsupervised Discovery of Persian Morphemes

Mohsen Arabsorkhi
Computer Science and Engineering Dept.,
Shiraz University,
Shiraz, Iran
marabsorkhi@cse.shirazu.ac.ir

Mehrnoush Shamsfard
Electrical and Computer Engineering Dept.,
Shahid Beheshti University,
Tehran, Iran
m-shams@sbu.ac.ir

Abstract

This paper reports the present results of a research on unsupervised Persian morpheme discovery. In this paper we present a method for discovering the morphemes of Persian language through automatic analysis of corpora. We utilized a Minimum Description Length (MDL) based algorithm with some improvements and applied it to Persian corpus. Our improvements include enhancing the cost function using some heuristics, preventing the split of high frequency chunks, exploiting penalty for first and last letters and distinguishing pre-parts and post-parts. Our improved approach has raised the precision, recall and f-measure of discovery by respectively %32, %17 and %23.

1 Introduction

According to linguistic theory, morphemes are considered to be the smallest meaning-bearing elements of a language. However, no adequate language-independent definition of the *word* as a unit has been agreed upon. If effective methods can be devised for the unsupervised discovery of morphemes, they could aid the formulation of a linguistic theory of morphology for a new language. The utilization of morphemes as basic representational units in a statistical language model instead of words seems a promising course [Creutz, 2004].

Many natural language processing tasks, including parsing, semantic modeling, information retrieval, and machine translation, frequently require a morphological analysis of the language at hand. The task of a morphological analyzer is to identify the lexeme, citation form, or inflection class of surface word forms in a language. It seems that even approximate automated morphological analysis would be beneficial for many NL applications dealing with large vocabularies (e.g. text retrieval applications). On the other hand, the construction of a comprehensive morphological analyzer for a language based on linguistic theory requires a considerable amount of work by experts. This is both slow and expensive and therefore not applicable to all languages. Consequently, it is important to develop methods that are able to *discover* and *induce* morphology for a language based on unsupervised analysis of large amounts of data.

Persian is the most-spoken of the modern Iranian languages, which, according to traditional classification, with the Indo-Aryan language constitute the Indo-Iranian group within the Satem branch of the Indo-European family. Persian is written right-to-left in the Arabic alphabet with a few modifications. Three of 32 Persian letters do double duty in representing both consonant and vowels: /h/, /v/, /y/, doubling, as /e/ (word finally), /u/, and /I/ respectively [Mahootian 97]. Persian morphology is an affixal system consisting mainly of suffixes and a few prefixes. The nominal paradigm consists of a relatively small number of affixes [Megerdoomian 2000]. The verbal inflectional system is quite regular and can be obtained by the combination of prefixes, stems, inflections and auxiliaries. Persian morphologically is a powerful language and there are a lot of morphological rules in it. For example we can derive more than 200 words from the stem of the verb “raftan” (to go). Table 1 shows some morphological rules and table 2 illustrates some inflections and derivations as examples.

There is no morphological irregularity in Persian and all of the words are stems or derived words, except some imported foreign words, that are not compatible with Persian rules (such as irregular Arabic plural forms imported to Persian.)

simple past verb	past stem + identifier
continuous present verb	Mi+present stem+identifier
Noun	present stem + (y)eš

Table 1. Some Persian morphological rules.

POS	Persian	Translation
Verb Infinitive	Negaštæn	to write
Present Verb Stem	Negar	Write
Past Verb Stem	Negašt	wrote
Continuous Present verb	mi-negar-æm	I am writing
Simple Past verb	negašt-æm	I wrote
Noun from verb	Negæreš	Writing

Table 2. Some example words.

2 Related Works

There are several approaches for inducing morphemes from text. Some of them are supervised and use some information about words such as part of speech (POS) tags, morphological rules, suffix list, lexicon, etc. Other approaches are unsupervised and use only raw corpus to extract morphemes. In this section we concentrate on some unsupervised methods as related works.

[Monson 2004] presents a framework for unsupervised induction of natural language morphology, wherein candidate suffixes are grouped into candidate inflection classes, which are then placed in a lattice structure. With similar arranged inflection classes placed near one candidate in the lattice, it proposes this structure to be an ideal search space in which to isolate the true inflection classes of a language. [Schone and Jurafsky 2000] presents an unsupervised model in which knowledge-free distributional cues are combined orthography-based with information automatically extracted from semantic word cooccurrence patterns in the input corpus.

Word induction from natural language text without word boundaries is also studied in [Deligne and Bimtol 1997], where MDL- based model optimization measures are used. Viterbi or the forward- backward algorithm (an EM algorithm) is used for improving the segmentation of the corpus. Some of the approaches remove spaces from text and try to identify word boundaries utilizing e.g. entropy- based measures, as in [Zellig and Harris, 1967; Redlich, 1993].

[Brent, 1999] presents a general, modular probabilistic model structure for word discovery. He uses a minimum representation length criterion for model optimization and applies an incremental, greedy search algorithm which is suitable for on- line learning such that children might employ.

[Baroni, et al. 2002] proposes an algorithm that takes an unannotated corpus as its input, and a ranked list of probable returning related pairs as its output. It discovers related pairs by looking morphologically for pairs that are both orthographically and semantically similar.

[Goldsmith 2001] concentrates on stem+suffix-languages, in particular Indo-European languages, and produces output that would match as closely as possible with the analysis given by a human morphologist. He further assumes that stems form groups that he calls *signature*s, and each signature shares a set of possible affixes. He applies an MDL criterion for model optimization.

3 Inducing Persian Morphemes

Our task is to find the correct segmentation of the source text into morphemes while we don’t have any information about words or any structural rules to make them. So we use an algorithm that works based on minimization of some heuristic cost function. Our approach is based on a variation of MDL model and contains some modifications to adopt it for Persian and improve the results especially for this language.

Minimum Description Length (MDL) analysis is based on information theory [Rissanen 1989]. Given a corpus, an MDL model defines a description length of the corpus. Given a probabilistic model of the corpus, the description length is the sum of the most compact statement of the model expressible in some universal language of algorithms, plus the length of the optimal compression of the corpus, when we use the probabilistic model to compress the data. The length of the optimal compression of the corpus is the base 2 logarithm of the reciprocal of the probability assigned to the corpus by the model. Since we are concerned with morphological analysis, we will henceforth use the more specific term the morphology rather than model.

$$(1)\quad \begin{aligned} &Descriptio\ nLength\ (Corpus\ C, Model\ M) = \\ &-\log_2 p(M) - \log_2 p(C \mid M) \end{aligned}$$

MDL analysis proposes that the morphology M which minimizes the objective function in (1) is the best morphology of the corpus. Intuitively, the first term (the length of the model, in bits) expresses the conciseness of the morphology, giving us strong motivation to find the simplest possible morphology, while the second term expresses how well the model describes the corpus in question.

The method proposed at [Creutz 2002; 2004] is a derivation of MDL algorithm which we use as the basis of our approach. In this algorithm, each time a new word token is read from the input, different ways of segmenting it into morphs are evaluated, and the one with minimum cost is selected. First, the word as a whole is considered to be a morph and added to the morph list. Then, every possible splits of the word into two parts are evaluated. The algorithm selects the split (or no split) that yields the minimum total cost. In case of no split, the processing of the word is finished and the next word is read from input. Otherwise, the search for a split is performed recursively on the two segments. The order of splits can be represented as a binary tree for each word, where the leaves represent the morphs making up the word, and the tree structure describes the ordering of the splits.

During model search, an overall hierarchical data structure is used for keeping track of the current segmentation of every word type encountered so far. There is an occurrence counter field for each morph in morph list. The occurrence counts from segments flow down through the hierarchical structure, so that the count of a child always equals the sum of the counts of its parents. The occurrence counts of the leaf nodes are used for computing the relative frequencies of the morphs. To find out the morph sequence that a word consists of, we look up the chunk that is identical to the word, and trace the split indices recursively until we reach the leaves, which are the morphs. This algorithm was applied on Persian corpus and results were not satisfiable. So we gradually, applied some heuristic functions to get better results. Our approach contains (1) Utilizing a heuristic function to compute cost more precisely, (2) Using Threshold to prevent splitting high frequency chunks, (3) Exerting Penalty for first and last letters and (4) Distinguishing Pre-parts and post-parts.

After analyzing the results of the initial algorithm, we observed that the algorithm tries to split words into some morphemes to keep the cost minimum based on current morph list so recognized morphemes may prevent extracting new correct morphemes. Therefore we applied a new reward function to find the best splitting with respect to the next words. In fact our function (equation (2)) rewards to the morphemes that are used in next words frequently.

(2) $$RF = \{ freq(LP) * (len(LP) - 1) / WN \} + \{ freq(RP) * (len(RP) - 1) / WN \} * C$$

In which LP is the left part of word, RP is the right part of it, Len (p) is the length of part P (number of characters), freq(p) is the frequency of part P in corpus, WN is the number of words (corpus size) and C is a constant number.

In this cost function freq(LP)/WN can be interpreted as the probability of LP being a morph in the corpus. We use len(P) to increase the reward for long segments that are frequent and it is decreased by 1 to avoid mono-letter splitting. We found the parameter C empirically. Figure 1 shows the results of the algorithm for various amounts of C.

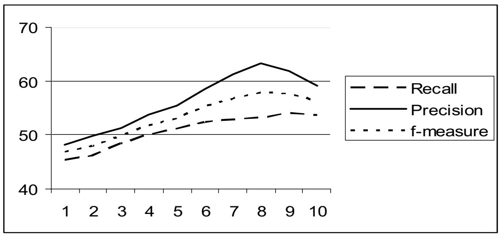

Figure 1. Algorithm results for various Cs.

Our experiments showed that the best value for C is 8. It means that RP is 8 times more important that LP. This may be because of the fact that Persian is written right-to-left and moreover most of affixes are suffixes.

The final cost function in our algorithm is shown in equation (3).

(3) $$F = \Delta E - RF$$

In which E is the description length, calculated in equation (1) and RF the cost function described in equation (2). Since RF values are in a limited range, they are large numbers (in comparison with other function values) in the first iterations, but after processing some words, cost function values will become large so that the RF is not significant any more. So we used the difference of cost function in two sequential processes (two iterations) instead of the cost function itself. In other words in our algorithm the cost function (E) is re-evaluated and replaced with its changes (ΔE). This improvement causes better splitting in some words such as the words shown in table 3. (Each word is shown by its written form in English alphabet : its pronunciation (its translation)).

word	Initial alg.	Improved alg.
šn: šen (sand)	šn	šn
šnva: šenæva (that can hear)	šn + va	šnv (hear) + a (subjective adjective sign)
mi-šnvm: mi-šenævæm (I hear)	mi (continuous tense sign) + šn + v + m	mi + šnv + m (first person pronoun)

Table 3. Comparing the results of the initial and improved algorithm.

We also used a frequency threshold T to avoid splitting words that are observed as a substring in other words. It means that in the current algorithm, for each word we first compute its frequency and it will be splitted just when it is used less than the threshold. Based on our experiments, the best value for T is 4.One of the most wrong splitting is mono-letter splitting which means that we split just the first or the last letter to be a morpheme. Our experiments show that the first letter splitting occurs more than the last letter. So we apply a penalty factor on splitting in these positions to avoid creating mono-letter morphemes.

Another improvement is that we distinguished between pre-part and post-part. So splitting based on observed morphemes will become more precise. In this process each morpheme that is observed at the left corner of a word, in the first splitting phase, is post-part and each of them at the right corner of a word is pre-part. Other morphemes are added to both pre and post-part lists.

4 Experimental Results

We applied improved algorithm on Persian corpus and observed significant improvements on our results. Our corpus contains about 4000 words from which 100 are selected randomly for tests. We split selected words to their morphemes both manually and automatically and computed precision and recall factors. For computing recall and precision, we numerated splitting positions and compared with the gold data. Precision is the number of correct splits divided to the total number of splits done and recall is the number of correct splits divided by total number of gold splits.

Our experiments showed that our approach results in increasing the recall measure from 45.53 to 53.19, the precision from 48.24 to 63.29 and f-measure from 46.91 to 57.80. Precision improvement is significantly more than recall. This has been predictable as we make algorithm to prevent unsure splitting. So usually done splits are correct whereas there are some necessary splitting that have not been done.

5 Conclusion

In this paper we proposed an improved approach for morpheme discovery from Persian texts. Our algorithm is an improvement of an existing algorithm based on MDL model. The improvements are done by adding some heuristic functions to the split procedure and also introducing new cost and reward functions. Experiments showed very good results obtained by our improvements.

The main problems for our experiments were the lack of good, safe and large corpora and also handling the foreign words which do not obey the morphological rules of Persian. Our proposed improvements are rarely language-dependent (such as right-to-left feature of Persian) and could be applied to other languages with a little customization. To extend the project we suppose to work on some probabilistic distribution functions which help to split words correctly. Moreover we plan to test our algorithm on large Persian and also English corpora.

References

Marco Baroni, Johannes Matiasek, Harald Trost 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity, *ACL Workshop on Morphological and Phonological Learning.*

Michael R. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery, *Machine Learning, 34*:71–105.

Mathias Creutz, Krista Lagus, 2002. Unsupervised discovery of morphemes. Workshop on Morphological and Phonological Learning of ACL'02, Philadelphia, Pennsylvania, USA, 21–30.

Mathias Creutz, Krista Lagus, 2004. Induction of a simple morphology for highly inflecting languages. *Proceedings of 7th Meeting of SIGPHON*, Barcelona. 43–51

S. Deligne and F. Bimbot. 1997. Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23:223–241.

John Goldsmith, 2001. Unsupervised learning of the morphology of a natural language, *Computational Linguistics,* 27(2): 153–198

Zellig. Harris, 1967. Morpheme Boundaries within Words: Report on a Computer Test. *Transformations and Discourse Analysis Papers, 73.*

Shahrzad Mahootian, 1997. Persian, Routledge.

Karine Megerdoomian, 2000 Persian Computational Morphology: A unification-based approach, NMSU, CLR, MCCS Report.

Christian Monson. 2004. A Framework for Unsupervised Natural Language Morphology Induction, *The Student Workshop at ACL-04.*

A. Norman Redlich. 1993. Redundancy reduction as a strategy for unsupervised learning. *Neural Computation*, 5:289–304.

Jorma Rissanen 1989, *Stochastic Complexity in Statistical Inquiry*, World Scientific.

P. Schone and D. Jurafsky. 2000. Knowldedge-free induction of morphology using latent semantic analysis, *Proceedings of the Conference on Computational Natural Language Learning.*