UNSUPERVISED SEGMENTATION OF WORDS INTO MORPHEMES

By
Orhan Eroglu
Hakan Kardes
Mustafa Torun

Submitted to the Department of Computer Engineering
in partial fulfilment of the requirements
for the degree of
Bachelor of Science
in
Computer Engineering

Boğaziçi University
April 2009

Table of Contents

List of Symbols

ASR, Automatic Speech Recognition
LSV, Letter Successor Variety
MAP, Maximum a Posteriori
MDL, Minimum Description Length
ML, Maximum Likelihood
OOV, Out Of Vocabulary
TUBITAK, Turkey Science and Technology Research Institution
UEKAE, National Research Institute of Electronics and Cryptology

ABSTRACT

Project Name : Unsupervised Segmentation of Words into Morphemes

Project Team : Orhan EROĞLU, Hakan KARDEŞ, Mustafa TORUN

Term : 2008/2009 2.Semester

Keywords : Unsupervised segmentation, Morfessor, language independent, vector representation, Turkish grammatical features, morpheme, morph, phone, natural language processing, agglutinative, phonetic features, unannotated text, raw text, Maximum a Posteriori, maximum likelihood, word frequency, lexicon, corpus.

Summary :

In this work we describe the application of the enhanced Morfessor algorithm with phonetic features to Turkish. Among the proposed segmentation approaches, the statistical Morfessor algorithm is a popular choice due to its ease of use for word segmentation. We aim to enhance baseline Morfessor algorithm with a basic phonetic knowledge of Turkish.

1. Introduction

1.1. Review of the Literature

The theory of morphology states that morphemes are considered to be the smallest meaning-bearing elements of language. Hence, any word form can be expressed as a combination of morphemes such as “ack-know-ledge-ment” for English or “gel-me-yecek-mi-sin” for Turkish.

One of the main objectives of natural language processing is to design an algorithm that segments words into morphemes in a language independent manner. There already exists open-source software, which performs the above objective in an acceptable manner, named ‘Morfessor1.0’ and implemented in Perl Script. It is a language-independent program which can be applied on any kind of languages (e.g. Concatenative languages, non-concatenative languages). However, since designers of Morfessor program have implemented it considering Finnish and English, and tested and evaluated the inputs and results also on those languages’ data and gold-standards, it is not as successful as English or Finnish, for Turkish.

Hence, some improvements can be done on our language.

1.2. Background of the Problem

Since Turkish is an agglutinative language with rich morphological features, it presents a considerable challenge for speech recognition systems. In order to obtain strong language model estimates, only the most common words of a language are used as the recognition vocabulary in state-of-the-art automatic speech recognition systems. This limitation on vocabulary results in high number of out-of-vocabulary (OOV) words, especially for agglutinative and highly inflectional languages, such as Turkish, Finnish, Estonian, Czech and Amharic. This situation directly results in a high word error rates. In this work we suggest improving Morfessor for Turkish without changing its language-independent character.

2. Statement of Problem

The morphological, phonetic features and vowel harmony of Turkish, makes the language processing objective, namely, 'segmentation of words into morphemes', more difficult with respect to other European languages. In order to address this problem, we aim to enhance the software Morfessor with a basic phonetic knowledge of Turkish without changing its language-independent feature.

We wanted to investigate new features that try to incorporate "oral" properties in the identification/selection of the sub-word units, in an attempt to take account of some specificities of spoken language. One of these new properties is based on the distinctive features specific to the Turkish phonemes. By giving a "phonemic" distance between two lexical units, word splits that result in the largest distances between sub-word units can be favored. The distinctive features are based on very general theoretical sound properties. Phonemes of a specific language are distinguishable with a small set of articulatory and acoustico-perceptive features, called distinctive features, such as voiced-unvoiced property or the place of articulation often corresponding to the point of constriction in the vocal tract.

3. Characteristics of Turkish

Turkish is a member of Altaic family of languages. The main characteristics of Turkish are the agglutinative morphology and the vowel harmony. These features distinguish Turkish as a challenging language for natural language processing and speech recognition applications. As a result of the agglutinative morphology, many new words can be derived from a single stem by addition of several suffixes. For instance,

"Aralattırmayabileceklerimkinden" is a single word which means
"Among those, which I cannot probably make anyone open out them"

	Open	Close	Open	Close
	Not rounded		Rounded	
Posteriors	a,[a]	ı,[W]	o,[o]	u,[u]
Anteriors	e,[e]	i,[i]	o,[ø]	¨u,[y]

Table 1: Turkish vowels with their [IPA] symbols

Labial	b [b], p [p], f [f], m [m], v [v]
Dental	d [d], t [t], s [s], z [z], n [n], l [l], r [r]
Palatal	c [dZ], c¸ [Ù], s¸ [S], j [Z], y[j]
Velar	g [g], k [k], v [w]
Uvular	h [h]

Table 2: Turkish consonants with their [IPA] symbols

Turkish is almost a phonetic language which consists of 29 graphemes, 8 vowels and 21 consonants. *Tables 1 and 2* give respectively the vowel and consonant inventories for Turkish. These properties will be used in deciding the phonetic features for Turkish word decompounding.

4. Work Done

4.1. Articles Related to Unsupervised Segmentation of Words

4.1.1. "Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor1.0" by Mathias Creutz and Krista Lagus

Mathias Creutz and Krista Lagus, in this article, describe the first public version of the Morfessor software, which is a program that takes as input a corpus of unannotated text and produces a segmentation of the word forms observed in the text.

The document contains user's instructions, the mathematical formulation of the model and a description of the search algorithm used. Furthermore, a few experiments on Finnish and English text corpora are reported in order to give the user some ideas of how to apply the program to their own data sets and how to evaluate the results.

With the help of this document, we learn how a word segmentation algorithm behaves in language-independent manner. Moreover, it is the document which is the starting point of our learning activities that will enlighten our way throughout the project. Our project proposal took shape after we study Mathias Creutz and Krista Lagus' article.

Additionally, we start studying Morfessor1.0 software under the light of information we learned from this article.

4.1.2. *"Unlimited Vocabulary Speech Recognition with Morph Language Models Applied to Finnish"* by Teemu Hirsimaki, Mathias Creutz, Vesa Siivola Mikko Kurimo, Sami Virpioja, Janne Pylkkönen

This article gives a detailed description of algorithm for segmenting a text corpus into statistical morphs, and compares the resulting language models with models based on two alternative methods. Moreover, it focuses on Viterbi algorithm which is applied in order to produce final morph segmentation of the words in the corpus. Given the morph lexicon and morph probabilities, this algorithm finds the segmentation of a word with the highest probability. With the help of this information, we recognize that Vitetrbi search can easily

provide segmentation for new word forms that were not present in the raw data.

4.1.3. "Unsupervised Discovery of Morphemes" by Mathias Creutz and Krista Lagus

This document presents two methods for unsupervised segmentation of words into morpheme-like units. One is "Recursive Segmentation and MDL Cost" and the other is "Sequential Segmentation and ML Cost."Thanks to this article, we learned the MDL and ML models of cost computation in detail. We learned about two different search approaches. One of them works by incrementally suggesting changes that could improve the cost function. 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. And the other utilizes batch learning where an EMlike (Expectation-Maximization) algorithm is used for optimizing the model.

4.1.4. "Two-step Approach to Unsupervised Morpheme Segmentation" by Stefan Bordag

The task is to find boundaries between morphemes bar any further analysis, "phoneme deletions, insertions or alternations that may ocur between or within morphemes." In order to address this problem, Stefan Bordag puts a two step algorithm forward. One step, based on a *letter successor variety* (LSV) revision, makes use of contextual information such as cooccurrences of words (the term 'word' will be used synonymously to 'word forms' throughout this paper) within sentences or next to each other. This second step of the algorithm is based on implementation of a classificator. The first step finds a boundary and each boundary trains the classificator. The classificator then, applied to an unanalysed word, marks the most probable prefix or suffix of that word. This article made us more skilled on segmentation approaches. Hence we had the chance of switch among different approaches for our different thoughts or possible suggestions.

4.1.5. "Unsupervised and Knowledge-free Morpheme Segmentation and Analysis" by Stefan Bordag

This article also defines two step approach for segmentation with an addition of a morphemic analysis based on contextual similarity. This provides knowledge about relatedness of the found morphs. For the boundary detection the challenge of increasing recall of found morphs while retaining a high precision is tackled by adding a compound splitter, iterating the LSV analysis and dividing the three classifiers into two distinctly applied classifiers. When we study on this article we started to think about a morphemic analysis based on contextual similarity, which will be an important part of our suggestion.

4.1.6. "Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency" by Mathias Creutz

This article presents an algorithm which is based on a new generative probabilistic model, which makes use of relevant prior information on the length and frequency distributions of morphs in a language. It is very similar even a pre-knowledge of the study "Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor1.0"

4.2. Contact with TUBITAK

The natural language processing study is one of the main objectives of the acoustic laboratory of TUBITAK/UEKAE. Some of the areas of their interest are

- Speech Intelligibility, Communicability and Quality Assessment;
- Speech Corpuses Design and Development;
- Speech and Speaker Recognition, Language and Accent Identification;
- Artificial Speech Synthesis;
- Speech and Video Coding

Each participants of our project group has attended a well-prepared internship program of acoustic laboratory of TUBITAK/UEKAE. We participated in data processing of their natural language processing study. Since we are familiar to their work, we decided to contact with them. Specifically, we are in touch with Coşkun MERMER and Ahmet Afşin AKIN. After meeting with them, thanks to their suggestions, we decided to drive our way on enhancing Morfessor software for Turkish. Especially Coşkun MERMER helped us for this issue, since he also studied on a similar topic.

Furthermore, Ahmet Afşin AKIN, who is the coder of Zemberek software, helped us for the milestones of the progress, segmentation of words into morphemes, in general. We keep in touch with TUBITAK/UEKAE and take their suggestions on our way.

Finally, when we are to finish our work, we wanted them to evaluate our results theoretically and also application-wise, several times. During evaluation, they gave us some hints about grammatical issues which we further tried to embed to our implementation.

4.3. Examination of Morfessor1.0 Software

The Morfessor1.0 software is one of our milestones for starting to our project. It is a program that takes a corpus of unannotated text as input and produces a segmentation of the word forms present in the text. Often, obtained segmentation resembles linguistic morpheme segmentation. Moreover, Morfessor is language-independent.

The data file of Morfessor is a word list with one word per line. Every word is preceded by a word count (frequency). Morfessor can be run on different sized data sets. The size of the data can vary from a few thousands of words to millions of words. There are some optional parameters of running the code:

- finish (float), which sets the convergence threshold for the search algorithm.
- rand (integer), which sets the random seed for the non-deterministic search.
- savememory, which is for reducing the memory consumption of the program.
- gammalendistr, a probability distribution function which is used for assigning prior probabilities to the lengths of morphs

- zipffreqdistr, a probability distribution function which is used for assigning prior probabilities to the frequencies (number of occurrences) of the morphs.
- trace (integer), which is for reporting the progress of the processing during the execution of the program.

In order to induce a model of language in an unsupervised manner from raw text, Maximum a Posteriori (MAP) estimate of the overall probability is used:

arg max P(M|*corpus*)=arg max P(*corpus*|M)*P(M)

P(M)=P(*lexicon, grammar*) ..... the probability of the model of language
P(*corpus*|M)…… the *maximum likelihood* (ML) estimate of the corpus, given the model of language.

These are for inducing a language model, which is a very important milestone of the software.

As the example above, there are lots of mathematical formulations for the Morfessor software. These are all explained in Mathias Creutz and Krista Lagus' article in detail. We just give the MAP estimate formulation for modeling, in order it to just give an idea about the whole undergoing mathematics.

It was very important for us to understand what is going on behind those formulations in order to be able to enhance the software for Turkish. We studied the code by three parts. One is cost computations of the software, one is search algorithms and the last one is segmentation part. During our studies on the code we gathered several times to incorporate our knowledge. We tried to understand the code line by line since the Perl script may do several things by just one line.

During and after our studies on the code, some suggestions, in our mind, started to take shape in order to contribute to Morfessor software.

4.4. Study on Turkish Grammatical Structure

After we contact with TUBITAK and meet Coşkun MERMER, we decided to drive our way on enhancing Morfessor1.0 for Turkish without violating language-independent character.

First, we studied Turkish grammar and its phonetic features. We recognize that, vowel harmony is a typical characteristic of Turkish. For example, according to one of the vowel harmony rules, a stem ending with a back/front vowel takes a suffix starting with a back/front vowel.

Language	Lexicon size (word types)	OOV(%)
English	65k	0.6
French	65k	1.2
Amharic	133k	6.9
Turkish	250k	6.5

Table 3: Table 3 Out-of-vocabulary rate (OOV) comparison for two rich morphology languages (Amharic and Turkish), and two languages that have a "less rich" morphology (English and French).

Table 3 gives lexicon sizes and OOV rates of systems developed at LIMSI for English and French, and for two morphologically rich languages, Amharic and Turkish. Nowadays, it is common practice to use lexicons comprised of at least 65k words and most state-of-the-art recognition system developers consider acceptable OOV rates to be under 1%. As shown in *Table 3* with 65k words the OOV rate for English is 0.6%, and is on the order of 1.2% for French. Using a 200k word lexicon can reduce the OOV rate to under 0.5% for French. For Amharic and Turkish, much higher OOV rates are observed, 6.5% and 6.9% respectively, with substantially larger lexicons. This difference is mainly due to the rich morphology of Amharic and Turkish, but is also accentuated by the lack of resources compared to English and French.

There are lots of such rules in Turkish grammar. We thought that if we embed some of suitable ones among those rules into the implementation, we may specify the code for Turkish. Hence, throughout our study on Turkish, we keep trying to recognize any specific feature suitable to embed and enhance the algorithm.

4.5. Running Morfessor1.0

While we study the code and read the related articles, we also run the program several times. We run it on Turkish data file which consist of about 580000 words and their frequencies. It was important for us to examine the outputs in order us to recognize any issue to be improved. Hence, after studying the structure of Turkish grammar, we studied on the output file for days, integrating our Turkish structure and Morfessor software knowledge. For a while we tried to notice any considerable defect on outputs that is suitable to settle using Turkish grammatical or phonetic characteristics. We divided the outputs in three parts in order us to be able to manage results easier. So, each of us investigated the output parts deeply. Meanwhile, we gathered to integrate our examination in some determined days.

4.6. Innovation

4.6.1. Modification to Morfessor Algorithm Regarding Phonetic Features

After all work we done, as mentioned above, we decided to enhance Morfessor for Turkish without demolishing its language-independent feature. Our prior study on Turkish grammar has made us to think about phonetic features.

All the properties used in the Morfessor program are based on written language and do not incorporate any “oral” properties that could be useful for ASR. We have recognized that vowel and consonant features for Turkish are generated by using the phonetic features shown in Tables 1 and 2. For instance, the properties associated with the vowel a[a] are open, not rounded and posterior. The vowel o[o] differs from the vowel a[a] as being rounded. By considering just these 3 properties, we have decided to represent the characteristics of vowels as vectors; i.e. a[a] and o[o] will be [1 0 1] and [1 1 1] respectively. Being rounded or not rounded is the only feature difference between these vowels. We try to find vector distance between two vowels of pairs of allomorphs.

This approach is an attempt in order to incorporate linguistic knowledge in decompounding process. A phone-based feature was added. This property aims to give an

estimation of the phonemic confusability between lexical units. It is theoretical and relies on some vowel features of the phones used in the language of study. For a particular morph, the smaller the feature value is, the greater the number of similar morphs (in terms of vowel features) there are in the lexicon. As for the other terms of the Morfessor algorithm, it takes the form of a probability.

4.6.2. Incorporating Phonetic Features to Baseline Morfessor 1.0

Morfessor is an iterative algorithm that given a corpus, proposes word segmentations found with an optimization criterion. The authors use the term of "morphemes" to name the sub-word units proposed by Morfessor, but they also use the neologism "morphs", since the splits are not always true morphemes in a linguistic sense. Finally, morphs can be either words or word splits.

During model training, the algorithm tries to iteratively maximize the following estimate:

$$\begin{aligned} M &= \underset{L}{\operatorname{argmax}}\ P(L|\text{corpus}) \\ &= \underset{L}{\operatorname{argmax}}\ \underbrace{P(\text{corpus}|L)}_{Likelihood}\ \underbrace{P(L)}_{A\ priori} \end{aligned} \tag{1}$$

where P(corpus|L) is the maximum likelihood estimate of the corpus given a lexicon L, based on the word frequencies, and P(L) is the *a priori* probability of the lexicon L, i.e., the probability of getting M distinct morphs m1, . . . ,mM as shown in equation 2. Properties used in the baseline version are morph frequency, morph string, and morph length, respectively denominated n(mk), s(m1) and l(m1) in equation 2. Our modifications, described in the following sections, affect the *a priori* properties used in equation 2.

$$\begin{aligned} P(\mathrm{L}) =& P(n(m_1),\ldots,n(m_M))P(s(m_1),\ldots,s(m_M))\ldots \\ & \ldots P(l(m_1),\ldots,l(m_M)) \end{aligned} \tag{2}$$

As it is common practice for this type of algorithm, probabilities are not multiplied as is, since they are often very small, but the negative log probabilities are summed. Maximizing the likelihood consists then in minimizing a sum of negative log probabilities, which can be seen as minimizing a cost function.

In both modes(), every word position is a potential candidate for split, and the algorithm explores all word substrings. Words can be split into various morphs, but words are not decompounded if splitting does not reduce the cost function value.

Based on these background, we have constructed some approaches and modifications:

1) ***End-of-Word Probability Modification:*** In the baseline Morfessor program, the character probabilities are static constants, calculated only once during model initialization, as the simple ratio of the number of occurrences of the character divided by the total number of characters in the corpus. These are independent of word position. To represent the word boundary, a space character is added to each lexical entry. The end-of-word probability is the probability of the space character, and has the same value for all words and morphs in the corpus.

   We propose replacing this static probability by the probability defined in *equation4*, to take the string context into account. P(l(m1), . . . , l(mM)) in equation 3 is replaced by P(l(m1), . . . , l(mM)). The word beginning symbol (WB) stands for the strings that begin a given word, from length zero to the word length itself. The probability that a word beginning WB is a morph, is defined as the ratio of the number of distinct letters L(WB) which can follow WB over the total number of distinct letters L. The division by L is not mandatory since it is a constant and thus does not influence the cost minimization, but it was kept for coherence, since the other quantities used in the algorithm are probabilities.

$$P_H(\mathrm{WB}) = L(\mathrm{WB})/L \quad (3)$$

2) ***Phonetic Features Modification:*** This modification is an attempt to incorporate linguistic knowledge in the decompounding process. A phone-based feature was added to the P(L) term of *equations 1 and 2*. This property aims to give an estimation of the phonemic confusability between lexical units.We define distance of two allomorphs, which is in the range [0, 1], as $D_{DF}(m_k)$, and

$$D_{DF}(m_k) = \prod_{J=1}^{N_k-1} D_{DF}(m_k, m_j) \quad (4)$$

$$D_{DF}(m_k, m_j) = \prod_{l=1}^{Vk} \frac{\Delta k_l, j_l}{C} \quad (5)$$

Where,

N_k : the total number of allomorphs which uses the same consonants as root

$\Delta k_l,j_l$: the number of different distinct features in the l^{th} vowel of morphs m_k and m_j

C: the total number of distinct features.

Equation 4 gives the definition for a morph m_k. The vowel features of its vowels are compared to the features of the vowels of all the other morphs that share the same consonantal root. The compared vowels have the same position in the morphs being compared.

The same definition is used for consonants; however in that case, the consonantal features of morphs that share the same "vocal root" are compared. For example, the two Turkish words with the phoneme transcriptions of [kola], [kolu], share the same [k, l] consonantal root. Thus the vowel features are compared. Both words have the same first vowel, which is ignored in the computation; otherwise the feature distance would be zero. Only the vowel pair [a,u] will have a contribution. The other possible vowel pairs [o, a] and [o, u] are not used since they involve vowels that have different word positions. In an analogous manner, if two words share the same "vocal root", then differences in the consonants can be computed.

The more distinct phonetic features two morphs have, the bigger the feature value is, and the smaller the associated "cost" (negative logarithm of PDF) is. This feature thus aims to favor word decompositions that give morphs which have distinct phonetic features compared to the other morphs.

To evaluate $\Delta k_l,j_l$, we have examined and used standard vowel and consonant feature tables for Turkish and Finnish, found in phonetics literature. The features used in this study concern vowels and consonants, and are given in the part 3 of the document.

Finally, as shown in *equation 6*, D_{DF} has been incorporated in P(L) as an additional term. *Equation 6* is our modified version of the original P(L) Morfessor formulation, given in *equation 2*. As for the other three properties (n, s, l), the property D_{DF} (m_k) is considered to be independent from the other morph feature values so that D_{DF} (m_k, . . . , m_k) = $\prod_{k=1}^{M} DDF\ (mk)$.

$$P(L) = P(n(m_1), \ldots, n(m_M))P(s(m_1), \ldots, s(m_M)) \ldots$$
$$\ldots P(l(m_1), \ldots, l(m_M))PDF\ (m_1, \ldots, m_M) \qquad (6)$$

5. Morpheme Segmentation Experiments

In the following, some differences between the tested Morfessor and MorfoBoun as well as the three tested languages are illustrated in the light of experimental results. The experiments were run on the datasets provided in the Morpho Challenge 2005. The Morfessor Baseline algorithm is entirely unsupervised and does not require that any parameters be set. The MorfoBoun algorithm has one parameter (the vowel and consonant features) that needs to be set to an appropriate value for optimal performance. This parameter value was optimized separately for each language on the small development sets (model segmentations) provided.

5.1. Results

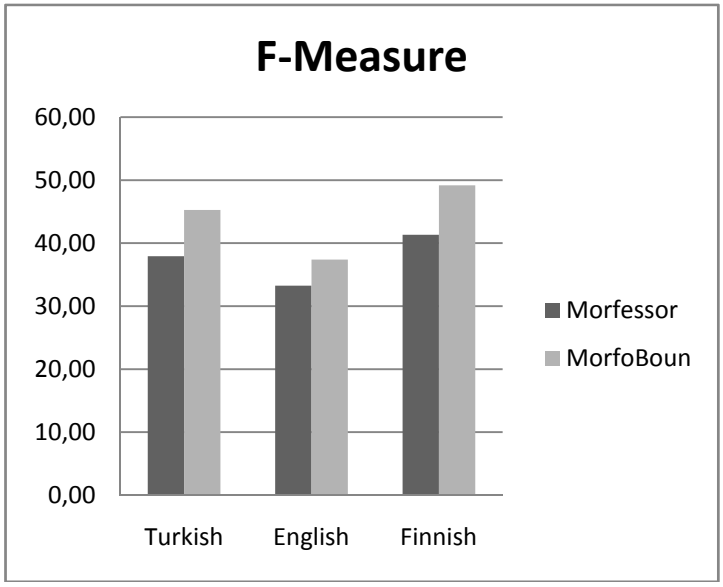

Figure1: F-measures computed for the placement of morpheme boundaries in relation to linguistic morpheme segmentations, obtained by the Morfessor and MorfoBoun on the three test languages.

The morpheme segmentation task of the competition is won by the participant achieving the highest *F-measure* of correctly placed morpheme boundaries. *Figure1* shows the F-measures of the two methods on the three tested languages. The F-measure is the harmonic mean of *precision* and *recall*. The precisions and recalls obtained by Morfessor are displayed in *Figures 2 and 3*, respectively. The results show that there are different tendencies for the English data, on the one hand, and the Finnish and Turkish data, on the other hand. For Finnish and Turkish, the context-dependent MorfoBoun model produce clear improvements

over the context-independent Morfessor splitting algorithm (with F-measures 7 – 8 points higher; *Figure1*). For English, the improvement is minor. The best F-measure obtained by Morfessor and MorfoBoun for all three languages is around 50%.

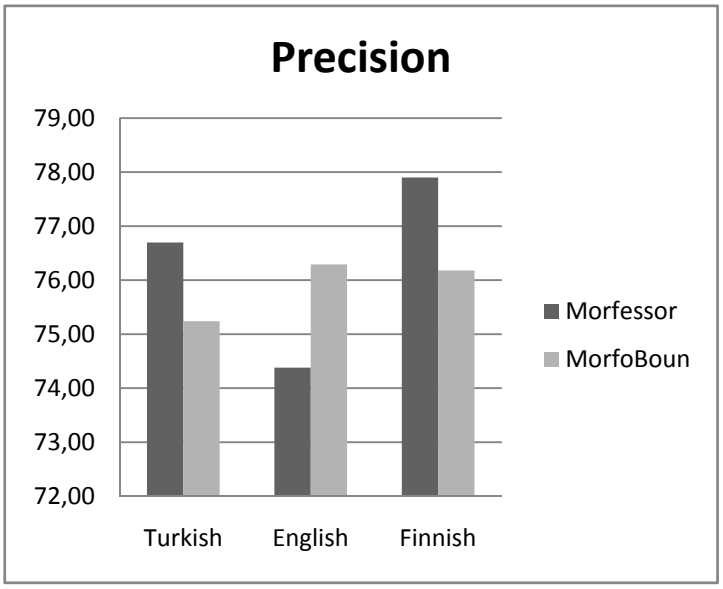

Figure2: Precision of the Morfessor and MorfoBoun methods on the three languages tested.

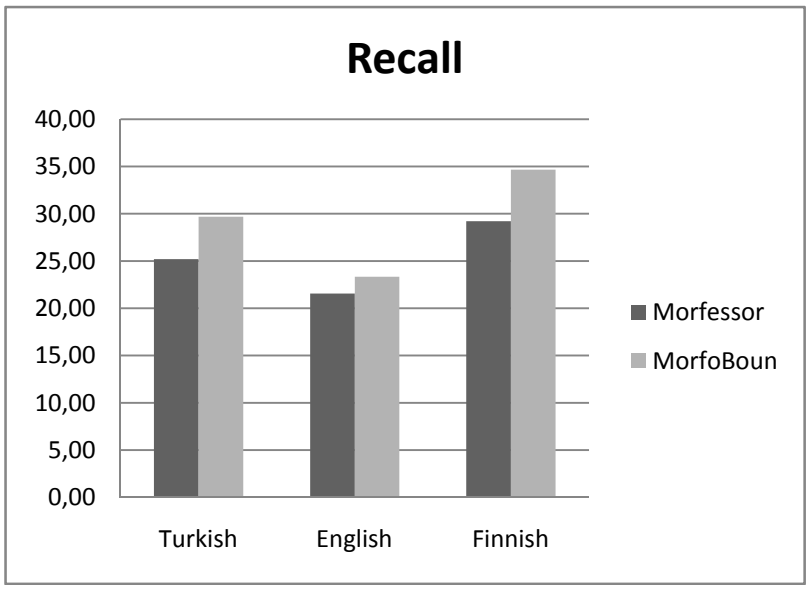

Figure3: Recall of the three Morfessor methods on the three languages tested.

The precision and recall plots in *Figures 2 and 3* provide more detailed information. For English, even though the F-measures of two algorithms are close to each other, the produced segmentations are very different. For Finnish and Turkish, the MorfoBoun model display a great improvement of recall in relation to the Morfessor method. This comes at the expense of lower precision, which is observed for Finnish and to a lesser degree on the

Turkish data. In order to better understand the differences observed in the results for the different languages, the output at various stages of the segmentation process has been studied for each of the Morfessor model and the MorfoBoun model. No obvious explanation has been found other than the difference in the morphological structures of the languages. Finnish and Turkish are predominantly agglutinative languages, in which words are formed through the concatenation of morphemes. The type/token ratio is high, i.e., the number of different word forms encountered in a piece of running text is relatively high. By contrast, word forming in English involves fewer morphemes. The type/token ratio is lower, and the proportion of frequently occurring word forms is higher.

6. Conclusion

The phonetic and vowel harmony features distinguish agglutinative languages as difficult languages for natural language processing, specifically for unsupervised segmentation.

In this report, we state our work; enhancing the main objective which is to design an algorithm that segments words into the smallest meaning-bearing units of language, morphemes, for Turkish and other agglutinative languages without violating language independence.

The work done in order to maintain our objective, ongoing study and the results of our experiments are stated in this paper.

7. References

[1] M. Creutz and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," Helsinki University of Technology, Publications in Computer and Information Science Report A81, March 2005.

[2] Teemu Hirsimaki, Mathias Creutz, Vesa Siivola Mikko Kurimo, Sami Virpioja, Janne Pylkkönen "Unlimited Vocabulary Speech Recognition with Morph Language Models Applied to Finnish," Helsinki University of Technology, Neural Networks Centre

[3] Mathias Creutz and Krista Lagus "Unsupervised Discovery of Morphemes" Helsinki University of Technology

[4] Harris, Z. "From Phoneme to Morpheme", Language **31**:190-222, 1996.

[5] Stefan Bordag "Two-step Approach to Unsupervised Morpheme Segmentation" University of Leipzig

[6] Thomas Pellegrini and Lori Lamel "Automatic Word Decompounding for ASR in a Morphologically Rich Language: Application to Amharic" LIMSI - CNRS

[7] Stefan Bordag "Unsupervised and Knowledge-free Morpheme Segmentation and Analysis" University of Leipzig

[8] Mathias Creutz "Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency" Helsinki University of Technology, Neural Networks Centre

[9] Micheal R. Brent "An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery"

[10] T. Pellegrini and L. Lamel, "Investigating Automatic Decomposition for ASR in Less Represented Languages," in *Proceedings of Interspeech*, Pittsburgh, 2006, pp. 285–288.

[11] P. Geutner, M. Finke, and A. Waibel, "Phonetic-distance-based hypothesis driven lexical adaptation for transcribing multlingual broadcast news," *Proceedings of ICSLP*, Sydney