Efficient Unsupervised Recursive Word Segmentation Using Minimum Description Length

Shlomo ARGAMON[a] **Navot AKIVA**[b] **Amihood AMIR**[b] **Oren KAPAH**[b]

[a]Illinois Institute of Technology, Dept. of Computer Science, Chicago, IL 60616, USA
argamon@iit.edu
[b]Bar-Ilan University, Dept. of Computer Science, Ramat Gan 52900, ISRAEL
{navot,amir,kapaho}@cs.biu.ac.il

Abstract

Automatic word segmentation is a basic requirement for unsupervised learning in morphological analysis. In this paper, we formulate a novel *recursive* method for minimum description length (MDL) word segmentation, whose basic operation is *resegmenting the corpus on a prefix* (equivalently, a suffix). We derive a *local* expression for the change in description length under resegmentation, i.e., one which depends only on properties of the specific prefix (not on the rest of the corpus). Such a formulation permits use of a new and efficient algorithm for greedy morphological segmentation of the corpus in a recursive manner. In particular, our method does not restrict words to be segmented only once, into a stem+affix form, as do many extant techniques. Early results for English and Turkish corpora are promising.

1 Introduction

Although computational morphological analyzers have existed for many years for a number of languages, there are still many languages for which no such analyzer exists, but for which there is an abundance of electronically-available text. Developing a morphological analyzer for a new language by hand can be costly and time-consuming, requiring a great deal of effort by highly-specialized experts. Supervised learning methods, on the other hand, require annotated data, which is often scarce or non-existent, and is also costly to develop. For this reason, there is increasing interest in *unsupervised* learning of morphology, in which unannotated text is analysed to find morphological structures. Even approximate unsupervised morphological analysis can be useful, as an aid to human annotators.

This paper addresses a key task for unsupervised morphological analysis: *word segmentation*, segmenting words into their most basic meaningful constituents (substrings), called *morphs* (orthographic realizations of morphemes). We adopt the minimum description length (MDL) approach to word segmentation, which has been shown to be effective in recent work (notably (Goldsmith, 2001) and (Brent et al., 1995)). The minimum description length principle (Barron et al., 1998) is an information-theoretic criterion to prefer that model for observed data which gives a minimal length coding of the observed data set (given the model) together with the model itself.

1.1 Our approach

Our approach in this paper is to better clarify the use of MDL for morphological segmentation by enabling direct use of a variety of MDL coding criteria in a general and efficient search algorithm. Issues of computational efficiency have been a bottleneck in work on unsupervised morphological analysis, leading to various approximations and heuristics being used. Our key contribution is to show how a *local* formulation of description length (DL) for word segmentation enables an efficient algorithm (based on pattern-matching methods) for greedy morphological segmentation of the corpus. We thus provide a search framework method which avoids some restrictions needed in previous work for efficiency. In particular, our method segments words in the corpus *recursively*, enabling multiple morphs to be extracted from a single word, rather than just allowing a single stem+affix pair for a given word, as in many previous approaches. For example, we might find the segmentation `inter+nation+al+ist`, whereas a single-boundary method would segment the word on just one of those boundaries.

This paper describes the first step in a larger research program; it's purpose is to show how to most efficiently recursively segment the words in a corpus based on an MDL criterion, rather than exhibit a full morphological analysis system. The procedure developed here is a component of a larger planned system, which will use semantic and structural information to correct word segmentation errors and will cluster morphological relations into productive paradigms.

1.2 Related work

Several systems for unsupervised learning of morphology have been developed over the last decade or so. Déjean (1998), extending ideas in Harris (1955), describes a system for finding the most frequent affixes in a language and identifying possible morpheme boundaries by frequency bounds on the number of possible characters following a given character sequence. Brent et al. (1995) give an information theoretic method for discovering meaningful affixes, which was later extended to enable a novel search algorithm based on a probabilistic word-generation model (Snover et al., 2002). Goldsmith (2001) gives a comprehensive heuristic algorithm for unsupervised morphological analysis, which uses an MDL criterion to segment words and find morphological paradigms (called *signatures*). Similarly, Creutz and Lagus (2002) use an MDL formulation for word segmentation. All of these approaches assume a stem+affix morphological paradigm.

Further, the above approaches only consider information in words' character sequences for improve morphological segmentation, and do not consider syntactic or semantic context. Schone and Jurafsky (2000) extend this by using latent semantic analysis (Dumais et al., 1988) to require that a proposed stem+affix split is sufficiently semantically similar to the stem before the split is accepted. A conceptually similar approach is taken by Baroni et al. (2002) who combine use of edit distance to measure orthographic similarity and mutual information to measure semantic similarity, to determine morphologically related word pairs.

2 Overview of the Approach

In this section we provide an overview of our approach to greedy construction of a set of morphs (a *dictionary*), using a minimal description length (MDL) criterion (Barron et al., 1998) (we present three alternative MDL-type criteria below, of varying levels of sophistication). The idea is to initialize a dictionary of *morphs* to the set of all word types in the corpus, and incrementally refine it by *resegmenting* affixes (either prefixes or suffixes) from the corpus. Resegmenting on a prefix p (depicted in Figure 1) means adding the prefix as a new morph, and removing it from all words where it occurs as a prefix. Some of the morphs thus created may already exist in the corpus (e.g., "cognition" in Fig. 1). We denote the set of morphs starting with p as V_p, and the set of *continuations* that follow p by S_p (i.e., $V_p = pS_p$). The number of occurrences of a morph m in the corpus (as currently segmented) is denoted by $C(m)$, and the number of tokens in the corpus with prefix p is denoted $B(p) = \sum_{v_k \in V_p} C(v_k)$.

Dictionary before	Dictionary after
relic	re
retire	lic
recognition	tire
relive	cognition
tire	live
cognition	farm
farm	

Figure 1: Illustration of resegmenting on the prefix `re-`. Note that $V_{\texttt{re}}$ ={relic, retire, recognition, relive}, and $S_{\texttt{re}}$ ={lic, tire, cognition, live}.

The algorithm examines all prefixes of current morphs in the dictionary as resegmentation candidates. The candidate $p*$ that would give the greatest decrease in description length upon resegmentation is chosen, and the corpus is then resegmented on $p*$. This is repeated until no candidate can decrease description length.

Key to this process is efficient resegmentation of the corpus, which entails incremental update of the description length change that each prefix p will give upon resegmentation, denoted ΔCODE_p (the change in the coding cost $\text{CODE}(M, \text{Data})$ for the corpus plus the model M). This is achieved in two ways. First, we develop (Sec. 3) expressions for ΔCODE_p which depend only on simple properties of p, V_p, and S_p, and their occurrences in the corpus. This *locality* property obviates the need to examine most of the corpus to determine ΔCODE_p. Second, we use a novel word/suffix indexing data structure which permits efficient resegmentation and update of the statistics on which ΔCODE_p depends (Sec. 4). Initial experimental results for the different models using our algorithm are given in Section 5.

3 Local Description Length Models

As we show below, the key to efficiency is deriving *local* expressions for the change in coding length that will be caused by resegmentation on a particular prefix p. That is, this coding length change, ΔCODE_p, should depend only on direct properties of p, those morphs $V_p = \{v_k = ps_k\}$ for which it is a prefix, and those strings $S_p = \{s_k | ps_k \in V_p\}$ (p's *continuations*). This enables us to efficiently maintain the necessary data about the corpus and to update it on resegmentation, avoiding costly scanning of the entire corpus on each iteration.

We now describe three description length models for word segmentation. First, we introduce local

description length via two simple models, and then give a derivation of a local expression for description length change for a more realistic description length measure.

3.1 Model 1: Dictionary count

Perhaps the simplest possible model is to find a segmentation which minimizes the number of morphs in the dictionary $\textsc{Code}^1(M, \text{Data}) = |M|$. Although the global minimum will almost always be the trivial solution where each morph is an individual letter, this trivial solution may be avoided by enforcing a minimal morph length (of 2, say). Furthermore, when implemented via a greedy prefix (or suffix) resegmenting algorithm, this measure gives surprisingly good results, as we show below.

Locality in this model is easily shown, as

$$\begin{aligned}\Delta\textsc{Code}_p^1(M) &= 1 + |S_p - M| - |V_p| \\ &= 1 - |S_p \cap M|\end{aligned}$$

since p is added to M as are all its continuations not currently in M, while each morph $v_k \in V_p$ is removed (being resegmented as the 2-morph sequence ps_k).

3.2 Model 1a: Adjusted count

We also found a heuristic modification of Model 1 to work well, based on the intuition that an affix with more continuations that are current morphs will be better, while to a lesser extent more continuations that are *not* current morphs indicates lower quality. This gives the local heuristic formula:

$$\Delta\textsc{Code}_p^{1a}(M) = 1 + |S_p - M| - \alpha|S_p \cap M|$$

where α is a tunable parameter determining the relative weights of the two factors.

3.3 Model 2: MDL

A more theoretically motivated model seeks to minimize the combined coding cost of the corpus and the dictionary (Barron et al., 1998):

$$\textsc{Code}^2(\text{Data}|M) + \textsc{Code}^2(M)$$

where we assume a minimal length code for the corpus based on the morphs in the dictionary[1].

The coding cost of the dictionary M is:

$$\begin{aligned}\textsc{Code}^2(M) &= \textsc{Code}^2(M) \\ &= b\textstyle\sum_{m\in M} \text{len}(m)\end{aligned}$$

where b is the number of bits needed to represent a character and $\text{len}(m)$ is the length of m in characters.

The coding cost $\textsc{Code}(\text{Data}|M)$ of the corpus given the dictionary is simply the total number of bits to encode the data using M's code:

$$\begin{aligned}&\textsc{Code}^2(\text{Data}|M) \\ &\quad= \textsc{Code}(M(\text{Data}) = M_{1\ldots N}) \\ &\quad= -\textstyle\sum_{i=1}^{N} \log P(m_i) \\ &\quad= -\textstyle\sum_{j=1}^{|M|} C(m^j) \log P(m^j) \\ &\quad= -\textstyle\sum_{j=1}^{|M|} C(m^j)(\log C(m^j) - \log N)\end{aligned}$$

where $M(\text{Data})$ is the corpus segmented according to M, N is the number of morph tokens in the segmented corpus, m_i is the ith morph token in that segmentation, $P(m)$ is the probability of morph m in the corpus estimated as $P(m) = C(m)/N$, $C(m)$ is the number of times morph m appears in the corpus, $|M|$ is the total number of morph types in M, and m^j is the jth morph type in the M.

Now suppose we wish to add a new morph to M by resegmenting on a prefix p from all morphs sharing that prefix, as above. First, consider the total change in cost for the dictionary. Note that the addition of the new morph p will cause an increase of $b\text{len}(p)$ bits to the total dictionary size. At the same time, each *new* morph $s \in S_p - M$ will add its coding cost $b\text{len}(s)$, while each *preexisting* morph $s' \in S_p \cap M$ will not change the dictionary length at all. Finally, each v_k is removed from the dictionary, giving a change of $-b\text{len}(v_k)$. The total change in coding cost for the dictionary by resegmenting on p is thus:

$$\begin{aligned}\Delta\textsc{Code}_p^2(M) = b\ (&\text{len}(p) \\ &+ \textstyle\sum_{s_k\in(S_p-M)} \text{len}(s_k) \\ &- \textstyle\sum_k \text{len}(v_k))\end{aligned}$$

Now consider the change in coding cost for the *corpus* after resegmentation. First, consider each preexisting morph type $m \notin V_p$, with the same count after resegmentation (since it does not contain p). The coding cost of each occurrence of m, however, will change, since the total number of tokens in the corpus will change. Thus the total cost change for such an m is:

$$\begin{aligned}&\Delta\textsc{Code}_p^2(\text{Data}|m \notin V_p) \\ &\quad= C(m)(\log P(m) - \log \hat{P}(m)) \\ &\quad= C(m)(\log C(m) - \log N - \log C(m) + \log \hat{N}) \\ &\quad= C(m)(\log \hat{N} - \log N) \\ &\quad= C(m)(\log(N + B(p)) - \log N)\end{aligned}$$

The total corpus cost change for unchanged morphs

[1] As is well known, MDL model estimation is equivalent to MAP estimation for appropriately chosen prior and conditional data distribution (Barron et al., 1998).

depends only on N and $B(p)$:

$$\begin{aligned}
&\Delta\textsc{Code}_p^2(\text{Data}|M - V_p) \\
&\quad = \textstyle\sum_{m \in M - V_p} C(m)(\log(N + B(p)) - \log N) \\
&\quad = (\textstyle\sum_{m \in M - V_p} C(m))(\log(N + B(p)) - \log N) \\
&\quad = (N - \textstyle\sum_{v_k} C(v_k))(\log(N + B(p)) - \log N) \\
&\quad = (N - B(p))(\log(N + B(p)) - \log N)
\end{aligned}$$

Now, consider explicitly each morph $v_k \in V_p$ which will be split after resegmentation. First, remove the code for each occurrence of v_k from the corpus coding: $C(v_k) \log P(v_k)$. Next, add a code for each occurrence of the new morph created by the prefix: $-C(v_k) \log \hat{P}(p)$, where $\hat{P}(p) = B(p)/(N + B(p))$ is the probability of morph p in the resegmented corpus. Finally, code the continuations s_k: $-C(v_k) \log \hat{P}(s_k)$ (where $\hat{P}(s_k) = \frac{\hat{C}(s_k)}{\hat{N}} = \frac{C(v_k)+C(s_k)}{\hat{N}}$ is the probability of the 'new' morph s_k). Putting this together, we have the corpus coding cost change for V_p (noting that $B(p) = \sum_{v_k} C(v_k)$):

$$\begin{aligned}
&\Delta\textsc{Code}_p^2(\text{Data}|V_p) \\
&\quad = \textstyle\sum_{v_k} C(v_k)[\ \log P(v_k) - \log \hat{P}(p) - \log \hat{P}(s_k)\ ] \\
&\quad = \textstyle\sum_{v_k} C(v_k)\ \ (\log C(v_k) - \log N \\
&\qquad\qquad\qquad + \log \hat{N} - \log B(p) \\
&\qquad\qquad\qquad + \log \hat{N} - \log \hat{C}(s_k)) \\
&\quad = \ \textstyle\sum_{v_k} C(v_k)(\log C(v_k) - \log \hat{C}(s_k)) \\
&\qquad + B(p)(2 \log \hat{N} - \log N) \\
&\qquad - B(p) \log B(p)
\end{aligned}$$

Thus the cost change for resegmenting on p is:

$$\begin{aligned}
&\Delta\textsc{Code}_p^2(M, \text{Data}) \\
&\quad = \Delta\textsc{Code}_p^2(M) + \Delta\textsc{Code}_p^2(\text{Data}|M) \\
&\quad = \ \Delta\textsc{Code}_p^2(M) + \Delta\textsc{Code}_p^2(\text{Data}|M - V_p) \\
&\qquad + \Delta\textsc{Code}_p^2(\text{Data}|V_p) \\
&\quad = \ b\left[\text{len}(p) + \textstyle\sum_{s_k \in (S_p - M)} \text{len}(s_k) - \sum_{v_k} \text{len}(v_k)\right] \\
&\qquad + \ (N - B(p))\ (\log(N + B(p)) - \log N) \\
&\qquad + \ \textstyle\sum_{v_k} C(v_k)(\log C(v_k) - \log \hat{C}(s_k)) \\
&\qquad\quad + B(p)(2 \log \hat{N} - \log N) \\
&\qquad\quad - B(p) \log B(p)
\end{aligned}$$

Note that all terms are local to the prefix p, its including morphs V_p and its continuations S_p. This will enable an efficient incremental algorithm for greedy segmentation of all words in the corpus, as described in the next section.

4 Efficient Greedy Prefix Search

The straightforward greedy algorithm schema for finding an approximately minimal cost dictionary is to repeatedly find the best prefix $p* = \arg\min_p \Delta\textsc{Code}_p(M, \text{Data})$ and resegment the corpus on $p*$, until no $p*$ exists with negative $\Delta\textsc{Code}$. However, the expense of passing over the entire corpus repeatedly would be prohibitive. Due to lack of space, we sketch here our method for caching corpus statistics in a pair of tries, in such a way that $\Delta\textsc{Code}_p$ can be easily computed for any prefix p, and such that the data structures can be efficiently updated when resegmenting on a prefix p. (A heap is also used for efficiently finding the best prefix.)

The main data structures consist of two tries. The first, which we term the *main suffix trie* (MST), is a suffix trie (Gusfield, 1997) for all the words in the corpus. Each node in the MST represents either the prefix of a current morph (initially, a word in the corpus), or the prefix of a *potential* morph (in case its preceding prefix gets segmented). Each such node is labeled with various statistics of its prefix p (denoted by the path to it from the root) and its suffixes, such as its prefix length $\text{len}(p)$, its count $B(p)$, the number of its continuations $|S_p|$, and the collective length of its continuations $\sum_{s_k \in S_p} \text{len}(s_k)$, as well as the current value of $\Delta\textsc{Code}_p(M, \text{Data})$ (computed from these statistics). Also, each node representing the end of an actual word in the corpus is marked as such.

The second trie, the *reversed prefix trie* (RPT), contains all the words in the corpus in reverse. Hence each node in the RPT corresponds to the suffix of a word in the corpus. We maintain a list of pointers at each node in the RPT to each node in the MST which has an identical suffix. This allows efficient access to all prefixes of a given string. Also, those nodes corresponding to a complete word in the corpus are marked.

Initial construction of the data structures can be done in time linear in the size of the corpus, using straightforward extensions of known suffix trie construction techniques (Gusfield, 1997). Finding the best prefix $p*$ can be done efficiently by storing pointers to all the prefixes in a heap, keyed by $\Delta\textsc{Code}_p$. To then remove all words prefixed by $p*$ and add all its continuations as new morphs (as well as $p*$ itself), proceed as follows, for each continuation s_k:

1. If s_k is marked in RPT, then it is a complete word, and only its count needs to be updated.
2. Otherwise
   (a) Mark s_k's node in MST as a complete word, and update its statistics
   (b) Add s_k^R to RPT and mark the corresponding nodes in MST as accepting stems.
3. Update the heap for the changed prefixes.

Prefixes		Suffixes	
re-	*ter-	-'s	*-at
un-	im-	-ing	-ate
in-	com-	-ed	-ive
de-	trans-	-es	-able
con-	sub-	-ly	-ment
dis-	*se-	-er	-or
pre-	en-	?-ers	-en
ex-	*pa-	-ion	?-ors
pro-	*pe-	?-ions	?-ings
over-	*mi-	-al	*-is

Figure 2: The first 20 English prefix and suffix morphs extracted from Reuters-21578 corpus using Model 1. Meaningless morphs are marked by '*'; nonminimal meaningful morphs by '?'.

Prefixes		Suffixes	
$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$
over-	un-	-'s	-'s
non-	over-	-ly	-ly
under-	non-	-ness	-ness
mis-	*der-	-ship	-ment
food-	dis-	?-ships	?-ments
stock-	mis-	?-ization	?-ized
feed-	out-	-ize	-ize
view-	inter-	?-ized	?-ization
work-	trans-	?-isation	?-izing
export-	re-	?-izing	?-isation
book-	super-	?-izes	?-ised
warn-	fore-	?-holders	-ise
borrow-	up-	?-izations	?-ising
depres-	down-	?-isations	?-ises
market-	tele-	-water	-ship
high-	stock-	?-ised	-men
narrow-	im-	-ise	?-ened
turn-	air-	?-ising	?-ening
trail-	euro-	?-ises	?-izes
steel-	mid-	?-iser	*-mental

Figure 3: The first 20 English prefix and suffix morphs extracted using Model 1a, as above.

The complexity for resegmenting on p is

$$O(\text{len}(p) + \sum_{s_k \in S_p} \text{len}(s_k) + \text{NSUF}(S_p) \log(|M|))$$

where $\text{NSUF}(S_p)$ is the number of different morphs in the previous dictionary that have a suffix in S_p (which need to be updated in the heap).

5 Experimental Results

In this section we give initial results for the above algorithm in English and Turkish, showing how meaningful morphs are extracted using different greedy MDL criteria. Recall that the models and algorithm described in this paper are intended as parts of a more comprehensive morphological analysis system, as we describe below in future work.

5.1 English

For evaluation in English, we used the standard Reuters-21578 corpus of news articles (comprising 1.7M word tokens and 32,811 unique words). For each of the 3 models described above, we extracted morphs either by resegmenting on prefixes or on suffixes (looking at the words reversed). When segmenting according to Models 1 and 2, a minimum prefix length of 2 was enforced, to improve morph quality (though not for suffixes, since in English there are some one-letter suffixes such as *-s*).

First, consider morphs found by Model 1 (Fig. 2). The prefix morphs found are surprisingly good for this simple model, with only one wrong in the first 15 extracted. That erroneous morph is `ter-`, which is part of `inter-`, however `in-` was extracted first; this kind of error could be ameliorated by a merging postprocessing step. The suffixes are similarly good, although oddly the system did not find `-s`, which caused it to find several composite morphs, such as `-ers` and `-ions`, which can get resegmented into their components (`-er+s` and `-ion+s`) later.

Model 1a also performs extremely well, for different values of α (we show just $\alpha = 1$ and $\alpha = 2$ in Fig. 3, for lack of space). Note that the morphs found by this model differ qualitatively from those found by Model 1, in that we get longer morphs more related to agglutination than to regular inflection patterns. This suggests that multiple statistical models should be used together to extract different facets of a language's morphological composition.

Finally, morphs from the more complex Model 2 are given in Fig. 4. As in Model 1a, Model 2 gives more agglutinative morphs than inflective morphs, and has a greater tendency to segment complex morphs (such as `-ification-`), which presumably will later be resegmented into their component parts (e.g., `-if+ic+at+ion`). This may enable construction of hierarchical models of morphological composition in the future.

5.2 Turkish

In addition to English, we tested the method's ability to extract meaningful morphs on a small corpus of Turkish texts from the Turkish Natural Language Processing Initiative (Oflazer, 2001), which consists of one foreign ministry press release, texts of two treaties, and three journal articles on translation. The corpus comprises 20,284 individual words, of which 5961 are unique. Turkish is a highly agglutinative language, hence a prime candidate for recursive morphological segmentation. Results for Models 1 and 2 are shown in Tables 5–8. Meaningful

Prefixes	
non-	rein-
bio-	over-
?disi-	*ine-
diss-	?interc-
video-	fluor-
financier-	wood-
quadr-	key-
*kl-	*kar-
weather-	vin-
*jas-	?kings-

Suffixes	
-'s	-ville
-town	-field
?-ification	?-ians
?-alize	?-alising
?-ically	?-ological
-tech	-wood
?-ioning	?-etic
?-sively	-point
?-nating	-tally
?-tational	*-uting

Figure 4: The first 20 English prefix and suffix morphs extracted using Model 2, as above, with $b = 8$.

p	Meaning
bahs-	talk (about)
terk-	leaving
redd-	refuse, rejected
zikr-	mention (someone)
bey-	Mr., sir
akt-	agree
haps-	(im)prison
birbirlerin-	one to another
şefin-	your chief
tedbirler-	precautions
birin-	somebody
hükümlerin-	your opinions
ülkesin-	his country
elimiz-	our hand
düzenlemelerin-	your arrangements
yerin-	your place
kendin-	yourself
devletler-	governments
biçimin-	your style
istediğim-	(thing) that I want

Figure 5: Turkish morphs segmented as prefixes using Model 1.

morphs were found using all models, with Model 2 finding longer morphs, as in English. We do note some issues with boundary letters for Model 2 prefixes, however.

6 Conclusions

We have given a firmer foundation for the use of minimal description length (MDL) criteria for morphological analysis by giving a novel local formulation of the change in description length (DL) upon resegmentation of the corpus on a prefix (or suffix),

p	Meaning	p	Meaning
-nin	of	-si	of
-nın	of	-ndan	from
-nı	your	-ları	*plural form*
-na	to your	-lar	*plural form*
-ler	*plural form*	-sı	of
-leri	*plural form*	-ların	your (*pl.*) (things)
-nda	at your	-lerine	to your (*pl.*) (things)
-ni	your	-ya	to
-lerin	your (things)	-lara	to (*pl.*)
-ki	that	-dir	is

Figure 6: Turkish morphs segmented as suffixes using Model 1.

p	Meaning	p	Meaning
hizmet(l)-	service	bahs-	mention
neden(l)-	reason	zih(in)-	memory
madd-	material	verg(i)-	tax
birbir-	one another	person(el)-	employee
belg-	document	biri-	one of
izlenim-	observation	verme-	giving
nitelik-	specification	vere(n)-	giver
en-	width	belirsi(z)-	unknown
dil-	language	bildirim-	announcement
bilg(i)-	knowledge	zikr-	mention

Figure 7: Turkish morphs segmented as prefixes using Model 2. Turkish letters in parentheses are not in the segmented morphs, though a better segmentation would have included them.

which enables an efficient algorithm for greedy construction of a morph dictionary using an MDL criterion. The algorithm we have devised is generic, in that it may easily be applied to any local description length model. Early results of our method, as evaluated by examination of the morphs it extracts, show high accuracy in finding meaningful morphs based solely on orthographic considerations; in fact, we find that Model 1, which depends only on the number of morphs in the dictionary (and not on frequencies in the corpus at all) gives surprisingly good results, though Model 2 may generally be preferable (more experiments on varied and larger corpora still remain to be run).

We see two immediate directions for future work. The first comprises direct improvements to the techniques presented here. Rather than segmenting prefixes and suffixes separately, the data structures and algorithms should be extended to segment both prefixes and suffixes in the current morph list, depending on which gives the best overall DL improvement. Related is the need to enable approximate matching of 'boundary' characters due to orthographic shifts such as -y to -i-, as well as incorporating other orthographic filters on possible morphs (such as requiring prefixes to contain a vowel). Another algorithmic extension will be to develop an

p	Meaning	p	Meaning
-isine	toward (someone)	-ilerine	to (*pl.*)
-nlerinin	of their (things)	-lemektedir	it does
-taki	which at	*-tik	
-isini	from, towards	-ilemez	cannot do
-yeti	to	-lerimizi	our things
-iyorsa	if (*pres.*)	-mun	from my
-ili	with	-mlar	(*plural*)
-likte	at (the place of)	-tmak	to
-'in	of	-unca	while
-imizden	from our	-lu	with

Figure 8: Turkish morphs segmented as suffixes using Model 2; tables as in Figure 5.

efficient beam-search algorithm (avoiding copying the entire data structure), which may improve accuracy over the current greedy search method. In addition, we will investigate the use of more sophisticated DL models, including, for example, semantic similarity between candidate affixes and stems, using the probability of occurrence of individual characters for coding, or using n-gram probabilities for coding the corpus as a sequence of morphs (instead of the unigram coding model used here and previously).

The second direction involves integrating the current algorithm into a larger system for more comprehensive morphological analysis. As noted above, due to the greedy nature of the search, a recombination step may be needed to 'glue' morphs that got incorrectly separated (such as un- and -der-). More fundamentally, we intend to use the algorithm presented here (with the above extensions) as a subroutine in a paradigm construction system along the lines of Goldsmith (2001). It seems likely that efficient and accurate MDL segmentation as we present here will enable more effective search through the space of possible morphological signatures.

Acknowledgements

Thanks to Moshe Fresko and Kagan Agun for help with the Turkish translations, as well as the anonymous reviewers for their comments.

References

M. Baroni, J. Matiasek, and H. Trost. 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL/SIGPHON-2002*, pages 48–57.

Andrew Barron, Jorma Rissanen, and Bin Yu. 1998. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October.

Michael R. Brent, Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering morphemic suffixes: A case study in minimum description length induction. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pages 21–30, Philadelphia.

Hervé Déjean. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In *Workshop on Paradigms and Grounding in Natural Language Learning*, pages 295–299, Adelaide.

S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. 1988. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. ACM Press.

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

Dan Gusfield. 1997. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press.

Zellig Harris. 1955. From phoneme to morpheme. *Language*, 31:190–222.

Kemal Oflazer. 2001. English Turkish aligned parallel corpora. Turkish Natural Language Processing Initiative, Bilkent University. http://www.nlp.cs.bilkent.edu.tr/Turklang/corpus/par-corpus/.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge free induction of morphology using latent semantic analysis. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 67–72, Lisbon.

Matthew Snover, Gaja Jarosz, and Michael Brent. 2002. Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In *ACL-2002 Workshop on Morphological and Phonological Learning*.