A Novel Evaluation Method for Morphological Segmentation

Javad Nouri, Roman Yangarber
Department of Computer Science
University of Helsinki, Finland
First.Last@cs.helsinki.fi

Abstract

Unsupervised learning of morphological segmentation of words in a language, based only on a large corpus of words, is a challenging task. Evaluation of the learned segmentations is a challenge in itself, due to the inherent ambiguity of the segmentation task. There is no way to posit unique "correct" segmentation for a set of data in an objective way. Two models may arrive at different ways of segmenting the data, which may nonetheless both be valid. Several evaluation methods have been proposed to date, but they do not insist on *consistency* of the evaluated model. We introduce a new evaluation methodology, which enforces correctness of segmentation boundaries while also assuring consistency of segmentation decisions across the corpus.

Keywords: Morphology, unsupervised learning, segmentation, evaluation, consistency.

1. Introduction

Over the last 15 years, a number of algorithms have been proposed for learning of the morphological structure of languages from raw data—large lists of words in the language (the corpus). This paper addresses the problem of evaluating the performance of such algorithms. Morphology learning is especially interesting for morphologically rich languages, where words can consist of many morphs, which are surface realizations of *morphemes*; a morpheme is a minimal unit, which carries meaning or syntactic function.

Ultimately, the problem of morphological analysis includes several aspects. One is identifying the *structure* of each word—which segments it consists of, and the types of segments. In many languages segments can be stems and affixes; affixes can be prefixes, suffixes, etc. Another aspect is identifying *allomorphy*, which involves linking different variants (allomorphs) of a given morpheme, which appear in different words, and typically depend on phonological context. For example, it is not trivial to discover that in English *try-ing* and *tri-ed* have the same stem morpheme.

It is well understood how one can build morphological analyzers for many languages. This is typically done by writing morphological grammars, based on dictionaries of morphemes for stems and affixes, and writing rules for combining the morphemes into words, e.g., (Koskenniemi, 1983). The morphological analyzer is then used to analyze any word in text, and is an essential low-level component in many tasks in natural language processing (NLP), such as parsing, machine translation, etc.

Unsupervised learning of morphology is the inverse problem: to infer the morphology of all words in the language, given only a large number of words—given no dictionaries, rules, etc. A number of algorithms have been proposed and their performance has been compared, e.g., in a series of shared-task competitions, the Morpho-Challenges, (Kurimo et al., 2010). The competitions provided annotated data, and tools for evaluation of performance.

The majority of algorithms for unsupervised learning of morphology (that we have reviewed), treat the *segmentation* of words into morphs either as the ultimate goal or as a first step on the way toward learning the "deeper" aspects of morphology, viz., allomorphy. Allomorphy (when it is treated) is seen by most as a second step, after a segmentation has been obtained. At the same time, some papers observe the difficulty of *evaluating* segmentations directly(Virpioja et al., 2011); we review some of these problems in this paper.

We agree that the ultimate goal of unsupervised learning of morphology is to learn *everything* about the morphological system. Our view is that segmentation is an interesting problem—whether as an end in itself, or as a first step toward learning deeper morphology—and as such it should be and can be evaluated *directly*.

The paper will present the details of new gold-standard annotation guidelines, and an algorithm to evaluate morphological segmentation of words based on this gold standard, focusing on the *consistency* of the segmentation.[1]

The paper is organized as follows: in Section 2. we define the problem of evaluation of morphological segmentation, in Section 3. we review methods for evaluation of morphology learning algorithms, in Section 4. the motivation for a new evaluation method is presented, followed by our approach to the evaluation in Section 5.. Experimental results are presented in Section 6. and we conclude in Section 7. with discussions and possible improvements.

2. Problem Definition

Our goal is to define a quantitative *measure of goodness* of a segmentation produced by a model, compared to a *gold-standard* (reference) segmentation. Several methods have been designed to evaluate segmentation algorithms, either *directly* using the resulting segmentations, or indirectly—by using the segmentation algorithm to perform another NLP task, such as speech recognition or information retrieval (Virpioja et al., 2011). While using the algorithm in a NLP task is useful for predicting the quality of the algorithm as used inside that application, our goal here is to devise a method to evaluate the segmentation *directly*.

It is important to mention that we address only the task of *unsupervised* word *segmentation*, i.e., decomposing words into their surface *morphs*, using a large corpus of words

[1] The evaluation software along with experimental data is available from `http://nlp.cs.helsinki.fi/morpho/`.

in the language; we do not address full morphological analysis which would produce a list of *morphemes* and/or morpheme classes for each word. The two problems are related, but not identical. We focus on evaluating algorithms that produce a segmentation for each word in the corpus, e.g., (Bernhard, 2008), Morfessor Categories-MAP (Creutz and Lagus, 2007), ParaMor (Monson et al., 2009), RALI-ANA (Lavallée and Langlais, 2010), MetaMorph (Tchoukalov et al., 2010), Morfessor Baseline (Creutz and Lagus, 2005), and others. As a consequence, we assume that if we concatenate the segmentations, the result is identical to the input word.

The problem can be viewed as evaluating how correctly the morpheme *boundaries* are placed within the word—compared to the gold-standard reference. For example, in English, *dogs* must be segmented as *dog+s* where + denotes a morph boundary between stem and suffix. Note that some languages (Arabic, German, etc.) use non-concatenative (e.g., *ablaut*) processes for inflection and derivation, which cannot be described by *segmentation*. We deal primarily with languages for which segmentation provides a good approximation to morphology—which includes many Indo-European languages, agglutinative languages from the Uralic family, Turkic, and many others.

3. Prior Work

We review methods for evaluating algorithms for automatic learning of morphology. These methods can be categorized as either *direct* approaches, which assign scores to models by examining the output of the model; or *indirect*, which evaluate the model as a component of the larger task. The evaluation is then done based on how much better the larger task performs when the model is used.

A review of evaluation methods for different morphological tasks can be found in (Virpioja et al., 2011), which also introduces several new variations of the existing algorithms. Beyond morphological segmentation, these tasks include *clustering of word forms* and *full morphological analysis*.

Direct evaluation methods are, in general, believed to better reflect the characteristics of the algorithm, while indirect methods complicate the evaluation task, since one needs to minimize the effect of other factors in the larger task (Virpioja et al., 2011).

Other evaluation methods, which evaluate more than segmentations, or require more output from the model than only morphological segmentations, are described in (Spiegler and Monson, 2010; Virpioja et al., 2011).

A widely used approach (Kurimo et al., 2006; Snyder and Barzilay, 2008; Poon et al., 2009) for evaluating segmentation of words is the boundary *precision*, *recall*, and *accuracy* (BPRA[2]), based on the number of reference boundaries found and missed. If a set of *correct* segmentation points for the given words could be provided, the segmentation task could be viewed as an information retrieval (IR) task, where the items to be retrieved are the segmentation boundaries. Then the standard IR evaluation measures precision(P), recall (R), and F-score (F) can be defined as:

$$P = \frac{|\{\text{gold standard boundaries}\} \cap \{\text{retrieved boundaries}\}|}{|\{\text{retrieved boundaries}\}|}$$
$$R = \frac{|\{\text{gold standard boundaries}\} \cap \{\text{retrieved boundaries}\}|}{|\{\text{gold standard boundaries}\}|}$$
$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{1}$$

Alternatively, using notation from binary classification, precision, recall, F-score and accuracy (A) can be computed as:

$$\begin{aligned} P &= \frac{tp}{tp + fp} \\ R &= \frac{tp}{tp + fn} \\ A &= \frac{tp + tn}{tp + tn + fp + fn} \end{aligned} \tag{2}$$

where tp, tn, fp, fn denote number of true positives, true negatives, false positives and false negatives, respectively. Positive (negative) means that the model found (did not find) a morph boundary annotated in the gold standard.

One evaluation scheme, based on the BPRA method and related to our work, can be found in (Creutz and Lindén, 2004; Creutz et al., 2005). To allow the model to choose one of several alternative reference segmentations, they define *fuzzy* boundaries; instead of a strict segmentation points in words, the model will receive credit if the segmentation boundary is placed anywhere in the area permitted by the reference set.

4. Motivation

The BPRA approach has been used in several studies (Virpioja et al., 2011); however, it has several important shortcomings. In most languages that have concatenative (or agglutinative) morphological processes, *it is not always clear* where the correct boundaries should be placed, because of the complexities in the morpho-phonology of the language. For example, in Turkish, consider the morphology of *ekmeği*, composed of *ekmek* ('bread') + *i* (accusative marker); *k* changes to *ğ* due to regular consonant mutation. One *model*—i.e., one learning algorithm—might segment this word as *ekmeğ+i*, considering *ekmeğ* an allomorph of *ekmek* and *i* as the accusative marker, while another model might segment it as *ekme+ği*, considering *ekme* as an allomorph of *ekmek*, and *ği* an allomorph of *i*. Similarly, the plural in English: analyzing *flies*, one model could posit that the plural marker is *-s* (as in *dog+s*) and the stem *fly* has an allomorph *flie-*. Another model might posit an allomorph for the stem *fli-* and an allomorph *-es* for the plural marker. Clearly, there is no way to insist that one of the models is "better" or more correct than the other. This makes specifying the gold-standard segmentation problematic.

One option is to enumerate all acceptable segmentations for a word and give credit to the model if the predicted segmentation matches any of the reference segmentations. Alternatively, *fuzzy* boundaries can be defined for words by

[2] Most earlier works do not include accuracy as one of the measures, however we will include it in this paper along with precision, recall, and F-score since it is a related and relevant measure.

marking an acceptable *region* where the boundary can be placed. If the model predicts a boundary anywhere within the region, it receives credit. Such an evaluation method is suggested in (Creutz and Lindén, 2004; Creutz et al., 2005) as part of the Hutmegs package, as an attempt to provide gold-standard segmentations for Finnish. The main problem with this approach is that it is too permissive: it ignores the question of consistency and allows the model to violate its own decisions. A good evaluation scheme should penalize the model for *inconsistent* behavior.

In our Turkish example, *ekmeği*: both of the models discussed are equally good, as long as they remain faithful to the decision they make *throughout the entire corpus*, i.e., the model that prefers *ekmeğ+i* should do so for all words where consonants mutate before vowels, such as *ekmeğ+e*, *eteğ+i*, *artığ+ı*, etc.; if the model violates this (its own) decision on some words, it should be penalized.

5. Gold Standard Annotation and Evaluation Algorithm

A *dilemma* is a (language-specific) situation where more than one segmentation may be acceptable according to the gold-standard annotation. A particular decision that a model makes in case of a dilemma, is called a *theory* that the model supports. For example, in English, the words *dogs* and *flies* may have gold-standard segmentations:

$$\begin{array}{ccccccccccccc} & & & & & & & & & X & & X & \\ d & o & g & + & s & \qquad & f & l & i & . & e & . & s \end{array}$$

Segmentation of *dogs* is unproblematic, the placement of the morph boundary is clear, marked with +. For *flies*, we wish to mark two alternative segmentations as acceptable. We do this by naming the dilemma, *X*, and specifying in the (language-specific) configuration that it has two acceptable theories, with boundary before **or** after *e* (not both). We indicate this by:

a. placing dots before and after *e*, each dot labeled by X,

b. specifying (in the configuration) in the definition of X that acceptable theories for X are 10 or 01.

This means that the word must be segmented as either *fli+es* or *flie+s*, respectively; 1 indicates the presence of a boundary, and 0 indicates absence of it.

The key point then is that if the evaluation corpus contains many similar words—*flies, tries, cries, supplies*, etc.—and they are all annotated similarly, then the model must segment *all* of these words according to theory 10 or according to theory 01—*consistently*. If the model is not consistent, it will be penalized. However, the penalty must be applied in such a way as to give it maximal benefit of the doubt. For instance, if the model segments above words as:

$$\begin{array}{r} flie+s \\ trie+s \\ crie+s \\ supplie+s \end{array}$$

it will get full credit for all four words (according to theory 01). However if the model outputs the following as its answer to the segmentation problem:

$$\begin{array}{r} fli+es \\ trie+s \\ crie+s \\ supplie+s \end{array}$$

although these segmentations are correct *individually*, the model, as a whole is behaving inconsistently and should not be given full credit. There are two possible scenarios here. One is to assume that for this dilemma (X) the correct theory is 01, meaning that the accepted segmentation pattern is "$\cdots ie+s$", in which case $\frac{3}{4}$ of the words will yield full credit. The other case is to consider the theory 10 as correct, "$\cdots i+es$", which will result in full credit for $\frac{1}{4}$ of the words. Since these two theories are both valid according to the configuration, at evaluation time, the evaluation algorithm must find which theory results in the highest score for the model. In this example the theory supported by the majority of the words should be chosen as the reference.[3]. This is what we mean by assuring *maximal benefit of the doubt*. By following this approach, the evaluation algorithm does not discriminate against any model, since it assures maximal possible score for every model. The theory that yields the maximal score for a dilemma is referred to as the "*decision*" that model makes, or the *theory* that it *supports*.

Many other kinds of dilemmas and theories can be defined. Each dilemma is marked with its own unique label in the gold standard annotation, along with the theories it admits. In our approach to evaluation of segmentations, given a dilemma, the model receives credit for selecting one of the theories that the gold standard accepts, in a consistent way. It is crucial to note that these dilemmas are independent from each other, i.e., supporting a theory in one dilemma does not restrict or encourage a theory of another dilemma. Given a gold-standard annotation, if we fix some theory for each dilemma, we can generate the set of unambiguous *correct* segmentations for these fixed theories. The evaluation algorithm then computes the boundary precision, recall, and accuracy (BPRA) to score the model's segmentation in the standard way.

The crucial feature of our method is that it identifies which theories are *preferred* by the system, given the segmentations that are being evaluated. A theory is preferred, if choosing it would yield the highest score for the system. Since all theories are equally valid, the evaluation algorithm must find the one that maximizes the overall score.

Computing this exhaustively would require listing all possible segmentation sets based on all possible theories for each dilemma. This would make the problem intractable when the number of dilemmas is large, since it will require going through $O(n^k)$ potential reference segmentations, where n is the number of dilemmas and k is the maximum number of theories for a dilemma.

The overall score that we aim to maximize for the system is accuracy, since maximizing precision or recall separately results in preferring under-segmentation or over-segmentation, respectively. One could maximize the F-score to balance recall and precision, but because of a com-

[3] Although this is true for this dilemma, we show in section 5.4. that majority does not always give maximal benefit of the doubt.

plicated and non-linear relation between F-score and the number of segmentation points, it is not obvious how it could help to perform the computations with reasonable time complexity. We show how maximizing accuracy, under the assumption of independence of the theories, allows us to produce a consistent evaluation score.

5.1. Dilemmas and Labels

As mentioned above, our focus in this evaluation method is on dilemmas that a human expert/annotator encounters while annotating the segmentations—alternative ways to segment a word into morphs that are legitimate, and only depend on the annotator's decisions about allomorphy. These dilemmas can depend on what morphemes are present in the word and in the context of what other morphemes they appear.

Each dilemma is identified by a unique label. There are labels of different *arities*, which will be explained briefly using examples.

5.2. Two-Way Dilemmas

This is the simplest type of possible dilemma. It is used when a human expert believes that placing a morph boundary between two symbols is optional, meaning that it is not absolutely necessary to segment it at that point. One example would be in cases where for historical reasons one might place a morph boundary between two potential morphs.[4] In other words, the morph pair is old enough ("ossified") to be legitimately considered as a single morph. One example in Turkish is *çıkmaz* (meaning 'stalemate', 'dead end') which is considered by many as a single morph, but could be segmented as *çık* (from *çıkmak*; 'to exit') and suffix *-maz* a form of negation in Turkish).

Another example is Finnish *arvo* (meaning 'value')[5]

```
   Y
arv.o
```

This means that the system under evaluation can decide either to segment at the points labeled Y or not to segment. Segmenting (placing a morph boundary) at this point corresponds to theory 1 and no boundary corresponds to theory 0; the valid theories for dilemma Y are 0 and 1. The practicality of this notation will become clear later, with labels of higher arity. This is declared in the gold-standard annotation, in the configuration file:

```
(Y 2 0 1)
```

The first element of the list is the symbol (label) that identifies the dilemma; meaning, Y is a dilemma with two possible theories/choices, of which (both!) 0 and 1 are acceptable theories.

5.3. Four-Way and Higher-arity Dilemmas

Sometimes in gold standard annotation we face dilemmas in which more than one potential boundary is involved; for example, *flies*, as above, will be annotated:

```
   X X
fli.e.s
```

It is important to note that since label X will be used in this type of dilemmas, it will *always* appear in pairs. Only one theory (ideally) should be selected as a segmentation decision by the system under evaluation, and for example if it chooses to segment at both points or neither, it should be penalized. To impose this kind of restriction, we use the following line in the label definition file:

```
(X 4 1 2)
```

the number 4 states that the dilemma X is a "4-way" dilemma, meaning that in total there are four possible theories—ways to perform the segmentation—that the system could follow; thus, there are $\log_2 4 = 2$ consecutive occurrences of X expected in the gold-standard words that contain this dilemma. The following numbers indicate which of the 4 possible segmentation theories are theoretically valid and can be chosen by the system without penalty. The numbers are decimal values of binary representations of the possible segmentation configurations, designating the presence of a morph boundary by 1 and lack of boundaries by 0. This is depicted in Table 1.

possible segmentation	**comparison to gold standard**	**numeric representation**
flie+s	fli.e.s fli e+s	$(01)_2 = 1$
fli+es	fli.e.s fli+e s	$(10)_2 = 2$

Table 1: Valid segmentation configurations for label X and word fli.e.s, with decimal representation. Segmenting (placing a morph boundary) at a point corresponds to 1 and not segmenting corresponds to 0.

Of course there can be such dilemmas where the annotator decides it is also acceptable to segment at *both* locations, or neither. In such a case, corresponding numbers $0 = (00)_2$ and $3 = (11)_2$ can be added to the list of allowable theories. Finally, this notation can be extended to dilemmas of higher order. For example the following configuration line would define a label for an 8-way dilemma (corresponding to 3 possible segmentation locations, hence, 3 consecutive labels in gold-standard annotation):

```
(C 8 3 5 6)
```

The list (3 5 6) indicates that the *valid theories* are those which have exactly two (out of the three) segmentation points selected.

5.4. Performance Measure of Segmentations

Given a set of our gold standard segmentations, computing the BPRA measures is straightforward. Using the annotation method described above, we can choose an arbitrary theory for each dilemma, and generate the fixed gold-standard segmentations based on this set of theories. We

[4] Or for any other reason.

[5] In this example, it is not quite clear how to deal with the "*o*". Although there exists a derivational suffix "*-o*" which is used to form nouns out of verbs, not nouns.

want to give the model under evaluation the freedom to choose any of the valid theories, but force it to be consistent throughout the corpus and make the same decisions in similar situations. The model will be penalized for breaking its own decision.

As discussed above, we need a way to determine, for each dilemma, which theory is *preferred* by the model. In the "ideal" situation, where the model chooses one theory for each dilemma and segments all examples according to the chosen theory, it is clear that the preferred theory is the one that the model is complying with, and there is no penalization.

We next consider the case where for a given dilemma, D, the model decides to segment instances according to one of the theories, d_1, but other instances according to another theory, d_2. Any of these theories could be chosen as the correct decision and the BPRA measures could be computed for them, but our goal is to give the model *maximum* benefit of the doubt. The theory that has been chosen most often throughout the data-set might seem to be the theory that should be preferred. Although this is true for one-way dilemmas, it is not always the case. The next example clarifies this:

Gold standard	Model response
Z Z 1. abc.d.e	abc d e
Z Z 2. fgh.i.j	fgh i j
Z Z 3. klm.n.o	klm n o
Z Z 4. pqr.s.t	pqr s+t
Z Z 5. uvw.x.y	uvw x+y
Z Z 6. zab.c.d	zab+c+d
Z Z 7. efg.h.i	efg+h+i

Figure 1: Left column: minimal example gold-standard annotation, with 7 instances of the 4-way dilemma Z. Right column: actual segmentations produced by a model to be evaluated using the gold standard; segmentation points are shown with +; blanks mark potential relevant segmentation points that were *not* segmented by the model (shown as blanks to help visualization).

The annotated gold standard for label (X 4 0 1 2 3) along with an example of a model's response is shown in Figure 1. As easily seen, the first three cases are segmented according to theory 00, the next two according to theory 01, and the last two according to theory 11; no word is segmented according to theory 10. It might seem that theory 00 is preferred by this particular model, but a closer look will falsify this. Let us compute the scores for each theory, assuming it is the preferred one. These score are showed in Table 2.

There are 7 words of length 5, so each has 4 potential segmentation boundaries. In total, there are 28 possible boundaries; 14 of these boundaries, which are not related to dilemma Z, are correctly left unsegmented by the model, regardless of the chosen theory. Depending on the preferred theory, some of the other boundaries are correct or incorrect. For instance if theory 00 is chosen, for each of the first three words, both boundaries relevant to Z will be counted as correct, so there will be 3 × 2 additional correct boundaries. For the next two words, only one of the relevant boundaries (the first one) will be counted as correct, which will account for 2 (compare this to the number of *bits in common* between the preferred theory and the one supported by words 4,5). The last two words have been segmented incorrectly according to theory 00 of Z, so they will not contribute anything to the final accuracy score. Thus, the accuracy in case of preferring 00 will be $\frac{14+3\times2+2\times1+2\times0}{28} = \frac{22}{28} = 0.786$. The same process is followed for the second theory resulting in an accuracy score of 0.821. Although 00 has more supporters than 01, it results in a lower accuracy.

The reason is that for 2-way and higher-order dilemmas, when one theory is preferred, words that have been segmented according to other theories—i.e., which support other theories—may still contribute to the accuracy score. This is because the theories are not disjoint, but have segmentation points in common and choosing one theory, might partially help other theories as well. Due to these indirect contributions to the total score, sometimes a theory with fewer supporters can surpass another theory with more supporters.

Theory	# Supporters	P	R	F	A
00	**3**	0	1	0	0.786
01	2	0.67	0.57	**0.62**	**0.821**
10	0	0.33	0.29	0.31	0.679
11	2	1.00	0.43	0.60	0.714

Table 2: Evaluation measures for the minimal example.

Thus, to give the model maximum benefit of the doubt, we need to choose the theory that maximizes one of the evaluation measures, not the number of supporters. Precision and recall are not good choices for maximization, since maximizing one of them will trade off with the other, resulting in under-segmentation or over-segmentation. F-score and accuracy are the remaining options. In our method, we give the models maximal benefit of the doubt in terms of accuracy since it is computationally easier, as shown below.

For this purpose, we need a score for dilemmas which maximizes accuracy. Let us denote the *set* of dilemmas as $\mathcal{D}$ and a valuation—i.e., an assignment of theories t to dilemmas in $\mathcal{D}$—as $[D]_t$ and the accuracy score under valuation t with A_t. When computing accuracy, the denominator $tp + tn + fp + fn$ is the number of possible boundaries in the gold-standard data, which is constant and independent of the chosen theories. More precisely, it is exactly $\sum_{w \in W_{gs}} (|w| - 1)$ where W_{gs} is the set of gold standard words. Thus, we maximize the numerator and find:

$$\begin{aligned}\arg\max_{[[D]]_t} A_t &= \arg\max_{[[D]]_t} \frac{tp + tn}{tp + tn + fp + fn} \\ &= \arg\max_{[[D]]_t} (tp + tn)\end{aligned}$$

By definition, dilemmas are independent of each other, and selecting one does not influence the others. Additionally, the ambiguities in words do not overlap, and each boundary in the gold-standard words can be marked with a maximum of one label. Thus, we can maximize the sum of true positives and true negatives *separately for each dilemma*, and solve the problem by finding the preferred theory for one dilemma at a time.

		Contribution to $tp + tn$			
		00	01	10	11
Candidate theory	00	2	1	1	0
	01	1	2	0	1
	10	1	0	2	1
	11	0	1	1	2

Table 3: Contribution of each theory to $tp + tn$ score for all candidate theories, for every instance of the dilemma in the data. Each cell shows how many correct segmentation boundaries will be contributed to the accuracy score if the theory corresponding to the row is preferred, and the theory corresponding to the column is supported by the word.

To do this, for each dilemma, we create a "contribution" table similar to Table 3. Each row corresponds to one candidate theory; given that the theory corresponding to the row is the preferred one, the table shows how many correct boundaries would be contributed by supporters of theories in the columns. Since the dilemma in the Table is a 4-way dilemma, for each instance two boundaries should be examined. For any instance, if the preferred theory is the same as the supported theory, there are two correct boundaries. If a different theory than the preferred one is selected, depending on how many segmentation decisions they have *in common*, it will contribute 1 point or none to the numerator. The number of common segmentation decisions is given by the number of bits the binary representations of the two theories have in common.
If the number of supporters of each theory is counted (denoted by n_t for theory t), the result can be tabulated similarly to Table 4. The sum in each row shows what the contribution to the numerator will be if the corresponding candidate theory the preferred one. The task is now straightforward: select the theory for which the sum is maximized.

Another potential measure to maximize benefit of the doubt is the F-score. However, it is more complex than the accuracy, since the denominator is not constant with respect to the chosen theories for dilemmas, thus it is not clear how one should find the preferred theory sets for each dilemma at a time without enumerating all possible valuations for dilemmas.

		Contribution to $tp + tn$			
		00	01	10	11
Candidate theory	00	$2n_0$	$+\ n_1$	$+\ n_2$	$+\ 0$
	01	n_0	$+\ 2n_1$	$+\ 0$	$+\ n_3$
	10	n_0	$+\ 0$	$+\ 2n_2$	$+\ n_3$
	11	0	$+\ n_1$	$+\ n_2$	$+\ 2n_3$

Table 4: Contribution of theories to $tp + tn$ for each candidate for the all occurrences of the dilemma. Each cell shows how many correct boundaries will be contributed to the accuracy score if the theory corresponding to the row is preferred by the whole data set and the theory corresponding to the column is supported by the word. n_t is the number of supporters of theory t in the data set. Decimal values representing theories are displayed for simplicity.

One potential problem with maximizing accuracy is the so-called *accuracy paradox*, which arises when the set of true positives is smaller than the set of false positives, and the model decides to unconditionally predict negative (i.e., leave all words unsegmented). The same also can happen when the true negative set is smaller than the true positive set and the model decides to segment at every potential segmentation point. In these two cases, the model can achieve a higher accuracy, while the F-score will be lower. The same principle of maximal benefit of the doubt can be applied to cases where it is important to avoid this paradox, however the optimization technique presented here for accuracy will not be useful, as is for that task and the task might have a larger time complexity. We choose accuracy for optimization in this paper, and will explore optimization of F-score as part of our future work.

5.5. Evaluation Algorithm

Given a set of segmented words (output of a model) and a gold standard annotation, as explained earlier, we take the following steps to evaluate the model:

1. Determine the set of decisions that the model has made, i.e., find the set of decisions that maximizes the benefit of the doubt as illustrated in Table 4.

2. Generate the reference segmentations of gold standard words based on the decisions.

3. Compare the response segmentations to the reference segmentations and calculate the true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). As earlier, "positive" means segmenting at a potential segmentation point (symbol boundaries).

4. Calculate the values of precision, recall, and F-score based on the computed values.

Most of the above steps are straightforward; we explain the details of step 1.
To achieve this, we first determine what decisions the model has made for each word. This is done via an alignment algorithm driven by Levenshtein distance.
An example of such an alignment is shown in Figure 2.
Thus, the overall schema of the evaluation algorithm can be summarized as follows:

Gold standard entry:

```
  P P I J
tek.e.m.i.s+i+s+sä
```

Response:

```
teke+misi+ssä
```

Alignment:

```
tek.e.m.i.s+i+s+sä
tek e+m i s i+s sä
```

Figure 2: An example of a gold standard to response alignment. Alignment of "`.e.`" to "` e+`" shows that this particular word prefers theory 01 for dilemma *P*. Similarly, alignment of "`.`" to "` `" shows a preference of theory 0 for dilemmas *I* and *J*.

1. For each dilemma D:
   (a) List all the words from the segmentation set that are relevant to D.
   (b) For each such word, find the supported theory using Levenshtein algorithm to align the word to the gold standard entry.
   (c) Count the number of supporters of each theory.
   (d) Determine which choice of the valid theories yields the maximum accuracy, similarly to Table 4.
2. Once decisions for all dilemmas are made, generate the final reference segmentations (one segmentation per word) based on the decisions.
3. Use BPRA to compute the scores.

6. Experiments

Experiments with evaluating consistent and inconsistent models using our method demonstrate the two key features:

1. If we have an *inconsistent* model, standard BPRA will give it a *higher* score than our method.
2. If we have two competing models, our method will give a higher score to the model that is *more consistent*.

For the experiments, we generated two segmentation sets from a gold standard annotated for a Finnish corpus, consisting of about 1000 words. Both sets contain valid segmentations for each word viewed *separately*, however one of the sets is inconsistent in choosing the alternative segmentation points. The consistent set is generated by fixing one of the valid theories for each dilemma, and segmenting the words according to the chosen theory. For generating the inconsistent set, for each word and for each dilemma relevant to that word, one of the valid theories is chosen at random, and the word is segmented according to the decisions. Other segmentation boundaries are left untouched. By definition, both segmentation sets will get the full score using the standard BPRA method, since all plausible segmentations are listed as alternatives in the gold standard, and any segmentation that is in that list will be accepted.

The evaluation results for the two segmentation sets with our new method are tabulated in Table 5. In addition to precision, recall, F-score and accuracy, several other statistics which are used to calculate the scores are also included. As it can be seen the consistent model gets full credit in all measures, whereas the penalization of the inconsistent model is reflected in the scores of Table 5.

	Consistent	**Inconsistent**
$\|G\|$	2617	2635
$\|S\|$	2617	2600
$\|G \cap S\|$	2617	2471
$tp + tn$	8212	7919
$tp + tn + fp + fn$	8212	8212
Precision	100%	95.04%
Recall	100%	93.78%
F-Score	100%	94.40%
Accuracy	100%	96.43%

Table 5: Evaluation results: consistent and inconsistent segmentation sets. $|G|$ is the number of segmentation points in the final reference set (based on gold standard and model's decisions), $|S|$ is the number of segmentation points in the model's response, $tp + tn$ denotes the number of matching boundaries, and $tp+tn+fp+fn$ is the number of potential segmentation boundaries.

7. Conclusion and Future Work

The features discussed in the previous sections distinguish our proposed evaluation method from methods used in prior work—where a model could not be required to segment data in a consistent fashion.

In conclusion, we note that "ambiguity" is inherent in morphological segmentation: it is impossible to posit a single segmentation that is "correct" in many cases in many languages. Thus the gold standard must provide *flexibility* to accommodate different theories of morphology. However, if two models—where one is consistent and one is inconsistent—receive equal score, that means that the evaluation method being used is not informative.

In addition to the above crucial features, our evaluation mark-up is easy to read and write by a human annotator, which is convenient for developing evaluation suites.

Our evaluation method, while it resolves the matter of consistency, still leaves certain problematic cases unresolved. We briefly discuss them here.

One case is what we consider a true *morphological ambiguity*: a word that can be legitimately segmented in more than one way. For example, Turkish *evini* (*ev*: 'home/house') can be analyzed as *ev*_N+POSS2+ACC or *ev*_N+POSS3+ACC. Ambiguity arises because +*in* is the 2nd person singular possessive marker (POSS2) and +*i* is 3rd person singular possessive marker, but the morphology requires an epenthetic +*n* after a POSS3 as a "buffer" consonant before certain suffixes are added—in this case, the accusative marker, which begins with a vowel.

This applies to all similar word endings, such as *+ını*, *+unu*, and *+ünü*; which one appears in an instance is determined by the rules of Turkish vowel harmony; this also occurs with other suffixes, such as dative, ablative, and other nominal cases.

We distinguish regular ambiguity from sporadic ambiguity. An ambiguity is regular if, as in the example above, there are many words that follow the same pattern, and have the same ambiguity. We plan to extend the current algorithm to address true ambiguities in future work.

Acknowledgments

Research supported in part by the Academy of Finland through the FinUgRevita Project, and the National Center of Excellence "Algorithmic Data Analysis" (ALGODAN).

8. Bibliographical References

Bernhard, D. (2008). Simple morpheme labelling in unsupervised morpheme analysis. In Carol Peters, et al., editors, *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pages 873–880. Springer Berlin Heidelberg.

Creutz, M. and Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Helsinki University of Technology.

Creutz, M. and Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, February.

Creutz, M. and Lindén, K. (2004). Morpheme segmentation gold standards for Finnish and English. Technical Report A77, Helsinki University of Technology.

Creutz, M., Lagus, K., Lindén, K., and Virpioja, S. (2005). Morfessor and Hutmegs: Unsupervised morpheme segmentation for highly-inflecting and compounding languages. In *Proceedings of the Second Baltic Conference on Human Language Technologies*, pages 356–370, Tallinn, Estonia.

Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production*. Ph.D. thesis, University of Helsinki, Finland.

Mikko Kurimo, et al., editors. (2006). *Proceedings of the PASCAL Challenge Workshop on unsupervised segmentation of words into morphemes*, Venice, Italy. PASCAL European Network of Excellence.

Kurimo, M., Virpioja, S., Turunen, V., and Lagus, K. (2010). Morpho-Challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden, July. Association for Computational Linguistics.

Lavallée, J.-F. and Langlais, P. (2010). Unsupervised morphological analysis by formal analogy. In Carol Peters, et al., editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 617–624. Springer Berlin Heidelberg.

Monson, C., Carbonell, J., Lavie, A., and Levin, L. (2009). ParaMor and Morpho Challenge 2008. In Carol Peters, et al., editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 967–974. Springer Berlin Heidelberg.

Poon, H., Cherry, C., and Toutanova, K. (2009). Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 209–217, Stroudsburg, PA, USA. Association for Computational Linguistics.

Snyder, B. and Barzilay, R. (2008). Unsupervised multilingual learning for morphological segmentation. In *ACL*, pages 737–745.

Spiegler, S. and Monson, C. (2010). EMMA: a novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1029–1037. Association for Computational Linguistics.

Tchoukalov, T., Monson, C., and Roark, B. (2010). Morphological analysis by multiple sequence alignment. In Carol Peters, et al., editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 666–673. Springer Berlin Heidelberg.

Virpioja, S., Turunen, V. T., Spiegler, S., Kohonen, O., and Kurimo, M. (2011). Empirical comparison of evaluation methods for unsupervised learning of morphology. *TAL*, 52(2):45–90.