Morpheme Segmentation for Kannada Standing on the Shoulder of Giants

Suma Bhat
Beckman Institute, University of Illinois, Urbana-Champaign, IL 61801, USA
spbhat2@illinois.edu

ABSTRACT

This paper studies the applicability of a set of state-of-the-art unsupervised morphological segmentation algorithms for the problem of morpheme boundary detection in Kannada, a resource-poor language with highly inflectional and agglutinative morphology. The choice of the algorithms for the experiment is based in part on their performance with highly inflected languages such as Finnish and Bengali (complex morphology similar to that of Kannada). When trained on a corpus of about 990K words, the best performing algorithm had an F-measure of 73% on a test set. The performance was better on a set of inflected nouns than on a set of inflected verbs. Key advantages of the algorithms conducive to efficient morphological analysis of Kannada were identified. An important by-product of this study is an empirical analysis of some aspects of vocabulary growth in Kannada based on the word frequency distribution of the words in the reference corpus.

KEYWORDS: Unsupervised morphological segmentation, Kannada language.

Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), pages 79–94,
COLING 2012, Mumbai, December 2012.

1 Introduction

With the ongoing quest for developing language processing techniques and tools for under-resourced languages there is an emerging need to study various aspects of these languages. Kannada, with nearly 70 million speakers is one of the 40 most spoken languages in the world. It is one of the scheduled languages of India and the official and administrative language of the state of Karnataka in South India. Its rich literary heritage has endowed the language with an immense written resource and efforts are currently underway to bring them to web scales. However, available computational tools for Kannada are only in their incipient stages. Simultaneously, there is an ever increasing number of internet users who are creating online materials in Kannada. As more information becomes available it becomes imperative to develop language processing tools that help us organize, search and understand information in Kannada. One such task is that of information retrieval and the time is ripe for developing efficient information retrieval algorithms for Kannada.

The role of stemming to improve retrieval effectiveness, particularly for highly inflected languages and monolingual retrieval has been well documented in (Larkey and Connell, 2003). Consequently, with the goal of developing a suitable stemmer for Kannada, the focus of this study is an exploration of the suitability of current state-of-the-art unsupervised morphological analyzers (studied for English and Finnish) for the task of morphological segmentation of words and eventual stemming in Kannada. In this sense, we see Kannada as a dwarf sitting on the shoulder of giants such as English and Finnish.

More specifically, we study the usefulness of a set of unsupervised learning of morphology (ULM) approaches towards addressing the problem of morpheme boundary analysis for Kannada. Our empirical study uses two corpora in Kannada and we compare the performance of the approaches with respect to addressing some of the challenges of Kannada morphology. A by-product of this study is an analysis of the word frequency distributions for the purpose of creating stop words in Kannada as also to quantify the productive processes of Kannada morphology. In this study, we restrict ourselves to studying morpheme segmentation noting the fact that eventual stemming is not a distant goal once we have reasonably segmented a word into its constituent morphemes.

The rest of this paper is organized as follows. In Section 2 we present a description in brief of the morphological analyzers for Kannada proposed thus far. Section 3 deals with an overview of the unsupervised methods we consider in this study and the challenges in Kannada morphological analysis. A description of our experiment is found in Section 4 with its subsections describing the corpora used, the evaluation methods and the results. Section 5 deals with the discussion of the results and error-analyses of the experiment. In Section 6 we analyze some aspects of Kannada with reference to its word frequency distribution. We conclude the paper with our conclusion and remarks in Section 7.

2 Related Prior Work

There is some amount of work done on morphological analysis in Kannada. Vikram and Urs (Vikram and Urs, 2007) present their prototypical morphological analyzer for Kannada based on finite state machines. There is a mention of its ability to handle 500 distinct noun and verb stems of Kannada.

Antony et al (Antony et al., 2010) outline the development of a paradigm-based morphological analyzer for Kannada verbs with the ability to handle compound verb morphology achieves a very competitive accuracy of 96.25% for Kannada verbs.

In (Ramasamy et al., 2011), Ramasamy et al. describe their implementation of a morphological analyzer and generator for Kannada. It is a rule-based finite state transducer with relevant morphological feature information of Kannada words and well defined morphophonemic (sandhi) rules governing word generation.

The morphological analyzer for Kannada described by Shambhavi et al. (Shastri, 2011) is a rule based approach which stores the possible paradigms for the roots available in a lexicon in a computationally efficient trie data structure. A given word is analyzed by matching it with the corresponding paradigm. It also performs a morphophonemic analysis of a word that does not reside in the lexicon by proceeding with suffix stripping and lexicon look-up iteratively. The developed system has the capability to can handle up to 3700 root words and around 88000 inflected forms.

Murthy (Murthy, 1999) describes a finite-state network based morphological analysis and generation system, MORPH, for Kannada. In essence, the system segregates the procedural and declarative processing between its two components, the network component - which has the capability of handing the analysis, and the process component - which has the capability of handing the morphophonemic decisions. The analysis proceeds in a series of affix stripping steps, from the input word to the root which is then checked against its stored lexicon. The performance of the system is reported to be 60 to 70% on general texts.

However, as of this study, the few attempts towards morphological analysis available in the literature have only marginal details about the studies, broad mentions about performance and no form of discussion or error-analysis via insights gained for word classes or methods that worked (or those that did not work) is available. What is clear from available literature is that all the above methods have pursued a rule-based and completely supervised approach. There have been no studies to understand the capabilities/limitations of an unsupervised approach to morphological segmentation in Kannada nor are other corpus-based analyses about Kannada available.

Consequently, this study is an attempt to fill the lacuna and has the following two goals - first, we explore the applicability of state-of-the-art models for unsupervised learning of morphology in Kannada. Here our focus is not only to analyze the performance in general but also to study their behavior handling the morphological complexities in Kannada as pointers in the direction of leveraging their results. Second, we perform an empirical analysis of one of the largest publicly available corpora in Kannada. To the extent of our knowledge, this is the first study exploring unsupervised techniques for Kannada and consequently, we expect it to drive future efforts towards developing further tools/algorithms that address the broader problem of stemming in Kannada.

3 Methods for Unsupervised Morphological Analysis

Being the most researched language in the natural language processing community, several unsupervised morphological analysis techniques have been implemented and studied for English. In the recent years, however, with the inclusion of other highly inflected European languages such as Finnish in the language processing to-do list, unsupervised methods are expanding to analyze morphologies more complex than that of English. For the purpose of our experiment, we focus our attention on three main approaches where the choice is based in part owing to their success with highly inflected languages such as Finnish and Bengali(complex morphology) and popularity in available literature. The methods that we will consider for our study are:

1. Goldsmith's method of unsupervised learning of morphology (Goldsmith, 2001),
2. Morfessor Categories-MAP (Creutz and Lagus, 2007), and,
3. High-Performance, Language-Independent Morphological Segmentation (Dasgupta and Ng, 2006, 2007).

3.1 Linguistica

Goldsmith's method of unsupervised learning of morphology (popularly known by the name of the tool, Linguistica[1] that implements this technique) is centered around the idea of minimum description length (MDL). Very broadly, MDL of the data is a combination of the length of morphology (in information theoretic terms) and length of compressed data (or compressed length of the corpus, given by probabilities derived from morphology). The learning heuristic then proceeds in steps of discovering basic candidate suffixes of the language using weighted mutual information, using these to find a set of suffixes, then using MDL to correct errors generated by heuristics. It starts with a corpus of unannotated text and produces a set of signatures, a signature being a pattern of affixes (prefixes or suffixes) that a stem takes (Goldsmith, 2001). An example suffix signature in English could be NULL.ed.ing.s, which combines with the stem *mark* to create the words *mark, marked, marking* and *marks*. In addition to this, the algorithm gives a list of stems, prefixes and suffixes with corresponding frequency information. This method will henceforth be referred to as Linguistica.

3.2 Morfessor Categories-MAP

Morfessor is an unsupervised method for the segmenting words into morpheme-like units. The idea behind the Morfessor model is, like Linguistica, to discover as compact a description of the data as possible. Substrings occurring frequently enough in several different word forms are proposed as morphs and the words are then represented as a concatenation of morphs. An optimal balance is sought between compactness of the morph lexicon versus the compactness of the representation of the corpus. For our study we use the most general of the currently available morfessor implementations of the generative probabilistic models designed for highly inflecting and compounding languages (Creutz and Lagus, 2007).

In Morfessor Categories, the segmentation of the corpus is modeled using a Hidden Markov Model (HMM) with transition probabilities between categories and emission probabilities of morphs from categories. Three categories are used: prefix, stem, and suffix and an additional non-morpheme (or noise) category. Some distributional properties of the morphs in a proposed segmentation of the corpus are used for determining category-to-morph emission probabilities. An important improvement in this model (compared to its predecessors) is that the morph lexicon contains hierarchical entries. That is, a morph can either consist of a string of letters (as in the previous models) or of two submorphs, which can recursively consist of submorphs. This in turn supports the agglutinative word structure of complex words. The Morfessor Categories algorithm has one parameter (the perplexity threshold b) that needs to be set to an appropriate value for optimal performance. Being a Maximum a Posteriori (MAP) model, an explicit probabilty is calculated for both the lexicon and the representation of the corpus

[1] Linguistica is publicly available at http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/

conditioned on the lexicon. Current versions of Morfessor attain an F-measure value of about 70% for the languages Turkish, Finnish and English. We will refer to this model by its family name, Morfessor.

3.3 Language-Independent Morphological Segmentation

The third algorithm we consider here (henceforth referred to as UnDivide from the name of the program accompanying the publication(Dasgupta and Ng, 2007)[2]) is an extension of Keshava and Pitler's algorithm (Keshava and Pitler, 2006) on language-independent techniques for morpheme induction. It is possibly the first to apply unsupervised learning to morphological parsing of an Indo-Aryan language (Dasgupta and Ng, 2006). At its core is the step for inducing morphemes using the heuristics in Keshava and Pitler's algorithm[3]. The induced morpheme list is then modified via three extensions -

- Employing a length-dependent threshold to prune the list of candidate affixes - here the rationale is that shorter morphemes (of length one or two) are likely to be more erroneous than their longer counterparts;
- Detecting composite suffixes via suffix strength and word-level similarity; and,
- Improving root induction via a simple but novel idea of using relative corpus frequency of the candidates.

The important features of this algorithm are its ability to move beyond one-slot morphology to handle words with multiple suffixes and the identification of inappropriate morpheme attachments. It achieves an F-score of 83.29% on Bengali.

The first two of the algorithms studied here are similar in that they have information theoretic optimization criteria and the heuristics are guided by probabilistic methods. The last of these is based on heuristics pertinent to word formations in general. The above algorithms seem attractive candidates for studying unsupervised morphological segmentation for Kannada owing to the fact that they have very few or no language dependent parameters and because of their reasonable performance with Bengali and Finnish (with morphological complexities such as that of Kannada).

3.4 Challenges to Morphological Analysis in Kannada

Kannada is one of the four major literary languages of the Dravidian family. Kannada is mainly an agglutinating language of the suffixing type. Nouns are marked for number and case and verbs are marked, in most cases, for agreement with the subject in number, gender and person. This makes Kannada a relatively free word order language. Morphologically rich languages such as Kannada, are characterized by a large number of morphemes in a single word, where morpheme boundaries are difficult to detect because they are fused together. In addition, rampant morphophonemic processes (sandhi), productive compounding and agglutinating morphology of inflectional and derivational suffixes (the latter mostly with words

[2] Its implementation is available at http://www.hlt.utdallas.edu/ sajib/.

[3] The key idea in this paper is to use words that appear as substrings of other words and transitional probabilities together to detect morpheme boundaries

of Sanskrit origin naturalized into Kannada) drive the prolific word formation processes of the language(Sridhar, 1990).

Another challenge at the level of word formations is that Kannada is diglossic - the formal or the literary variety differs significantly from the spoken (informal) or the colloquial variety. For example, the first person singular form of the verb 'tinnu' (eat) in the non-past tense is 'tinnuttEne' in the literary variety and 'tinnuttIni' (which gets further simplified to 'tiMtIni') in the spoken variety. Here we restrict ourselves to the analysis of the literary variety but it must be pointed out that informal written materials (including plays, short stories or humorous articles) can include a lot of spoken forms.

It may be worth noting here that effective language processing techniques for a language like Kannada cannot a rely on a purely rule-based or a purely stochastic approach, since the former demands subtle linguistic expertise and elaborate hand coding whereas the latter a large and diverse corpus. What is needed is an efficient combination of both approaches.

4 Experiments

4.1 Data

For purposes of experimentation, we use two corpora whose sizes are tabulated in Table 1.

1. A collection of stories for children, *dinakkoMdu kathe* written in Kannada [4] by one of the leading writers Dr. Anupama Niranjana. This corpus being a collection of children's stories, is informal in nature as a result of which, includes a widespread use of informal constructions. Although our focus in this study is on the formally constructed word forms, efforts needed to clean the data prevent us from excluding the informal constructions from the corpus.

2. The set of written documents in Kannada from the EMILLE-CIIL corpus (EMI).The documents comprise a collection of essays on diverse topics including commerce, leisure, social sciences and literature.

Corpus name	No. of word tokens	No. of word types
Story collection (**DINA**)	96370	24851
EMILLE (**EMIL**)	997012	210368

Table 1: Data size of Kannada corpora.

Vocabulary creation: We then preprocess each of these data sets (romanized) by tokenizing (we follow the standard tokenization of counting word tokens as units delimited by spaces) and removing punctuations and other unwanted character sequences. The remaining words are then used to create our vocabulary, which consists of 24851 word types for the story collection and 210368 for the EMILLE set. Unlike morphological analysis for many European languages, however, we do not take the conventional step of removing proper nouns from our word list, because we do not have a named entity identifier for Kannada.

[4] We would like to acknowledge the help of the members of Sriranga Digital Software Technologies - Prof. C. S. Yogananda and D. Shivashankar, in Srirangapatna, Karnataka, India, who made this corpus available.

Test set preparation: To create our test set, we first get a list of common words in the vocabularies of the two corpora. We then manually removed the proper nouns, informal forms for verbs *hyAge,naMge*, words with spelling mistakes and high frequency stop words from the common word list before performing hand-segmentation of the words. In the absence of a complete knowledge-based morphological parsing tool and a publicly available hand-tagged morphological database for Kannada, we had to annotate on a subset of the common word list for generating our test cases. The test set consists of 53 inflected noun forms and 50 verb forms amounting to a total of 103 words. The verb forms included are those with one of the several aspectual auxiliaries, e.g. *hArihOyitu*(also termed as vectors, (Sridhar, 1990)), as well as those with various tense, aspect and PNG markers. The noun forms are those with the form noun+case ending, or noun+plural+case.

Gold standard: We obtain the gold standard by segmenting the words in the test set and for the purpose of this study we are looking for a surface-level segmentation. That is, the segmented word form must contain exactly the same letters as the original, unsplit word. Thus there may be multiple surface-level segmentations for a given word. For instance, the inflected verb form *ODidaru* has three surface segmentations - OD idaru, OD id aru and ODi daru all considered valid for this study.

4.2 Evaluation of the Chosen Methods

We run the algorithms separately for the two corpora with their default parameters. We then evaluate the results by comparing the segmentation of the words in the test set with that in the gold segmentation. The algorithms were run in their default parameter settings since a large enough gold standard to tune the parameters of the algorithms was unavailable.

The evaluation is based on the placement of morpheme boundaries and follows the guidelines set by the Morphochallenge competition[5]. We use F-score to evaluate the performance of the segmentation algorithms on the test set. F-score is the harmonic mean of recall and precision, which are computed based on the placement of morpheme boundaries as below.

- Precision is the number of hits (H) divided by the sum of the number of hits and insertions (I): Precision = H/(H+I).
- Recall is the number of hits divided by the sum of the number of hits and deletions (D): Recall = H/(H+D).
- F-Measure is the harmonic mean of precision and recall: F-Measure = 2H/(2H+I+D).

Here a hit occurs for every correctly placed boundaries between morphemes, an insertion for every incorrect boundary between morphemes, and a deletion for every missed boundaries between morphemes - all with respect tot the gold segmentation.

In many cases, it is difficult to come up with one single correct morpheme segmentation. However, we will use the provided gold standard as the only correct answer. For some words, there are multiple interpretations in the gold standard. All of them are considered correct, and the alternative that provides the best alignment against the proposed segmentation is chosen.

[5] For details see http://research.ics.aalto.fi/events/morphochallenge2005/evaluation.shtml.

4.3 Results

Linguistica and UnDivide in their default settings exclude the least frequent words (those seen only once in the corpus) in their analysis. Consequently, we first run all the algorithms with words that occur at least twice in the corpus. The corresponding results are summarized in Table 2

	DINA			**EMIL**		
Algorithm	P	R	F	P	R	F
Linguistica	62.64	47.11	53.77	45.63	38.21	41.59
Morfessor-CatMAP	73.63	59.82	**66.01**	67.86	68.47	68.16
UnDivide	63.75	45.13	52.85	72.348	71.58	**71.96**

Table 2: Evaluation results (reported in terms of precision (P), recall (R) and F-score (F))

Based on the results, it appears that the UnDivide algorithm by far outperforms the other algorithms for the large data set (EMILLE), whereas Morfessor is the best performing algorithm for the small data set.

In order to assess the performance of the segmentation algorithm on specific word categories (noun/verb), we do the following. We split the test set into two - a set of inflected nouns and another that of inflected verbs and evaluate the performance for the 6 data set and algorithm combinations. As tabulated in Table 3, we observe that for inflected nouns, UnDivide emerges as the best performer for EMIL whereas, Morfessor emerges as the best performer for DINA. In the case of inflected verbs, however, we notice that UnDivide is the best performing algorithm for both the data sets.

	DINA			**EMIL**		
Algorithm	P	R	F	P	R	F
Linguistica	67.92	59.02	63.16	40.98	47.17	43.86
Morfessor-CatMAP	93.33	73.68	**82.35**	80.36	80.36	80.36
UnDivide	64.86	42.11	51.06	79.25	85.71	**82.35**

Table 3: Evaluation results for inflected nouns (reported in terms of precision (P), recall (R) and F-score (F)).

	DINA			**EMIL**		
Algorithm	P	R	F	P	R	F
Linguistica	55.26	35.00	42.86	44.00	35.48	39.29
Morfessor-CatMAP	54.35	45.45	49.50	55.36	55.36	55.36
UnDivide	62.79	47.37	**54.00**	63.41	55.32	**59.09**

Table 4: Evaluation results for inflected verbs (reported in terms of precision (P), recall (R) and F-score (F)).

Noting that several morphological variants of more frequent stems belong to the set of words occurring only once, we extend the input data to include them and use all words in the analysis. The resulting change in performance is shown in Table 5 where we compare the results using the trimmed data set (no singletons) with those obtained using the full data set. Here we would like to mention the fact that Morfessor showed a bigger change in performance compared to UnDivide[6] the other results are not tabulated. In both instances we notice an increased recall and hence an increase in the F-measure.

	Trimmed			Full		
Algorithm	P	R	F	P	R	F
Morfessor (EMIL)	67.86	68.47	68.16	67.18	80.00	73.03
Morfessor (DINA)	73.63	59.82	66.01	66.91	82.73	73.98
UnDivide(EMIL)	72.348	71.58	71.96	68.97	73.39	71.11
UnDivide(DINA)	63.75	45.13	52.85	63.75	45.13	52.85

Table 5: Evaluation results including all words in the corpus(reported in terms of precision (P), recall (R) and F-score (F)) for Morfessor and UnDivide. We notice an improved F-measure owing to an increase in recall in all the cases except for DINA with UnDivide. Linguistica did not show any change.

5 Discussion and Error Analysis

5.1 Effect of Data size

With the exception of Linguistica, we notice improved morpheme boundary detection (in terms of increased F-measure) with increase in data sizes. A plausible explanation is that with increased data sizes, more morphological variants are available permitting better stochastically motivated decisions. In the case of Linguistica, however, we noticed that only about 65% of the input data was analyzed (recall that the training data excludes words occurring only once) and so data sparsity with inadequate access to morphological variants was an obvious reason for poor performance.

The improved performance resulting from an increased recall (as seen from Table 5) can be explained as follows. With access to more inflected forms of the same stem, the algorithm has an improved ability to segment the morphemes. As an example with the segmentation by Morfessor, consider the form *mADuttA* which in the trimmed case, was not segmented but in the full case was segmented as *mADutt A*. Again, consider the case of *janarige* which in the trimmed case was not segmented, but in the full was segmented as *jana ri ge*. This has the potential to decrease the number of deletions and in turn cause a corresponding increase in the number of hits (refer back to Section 4.2 for the meaning of hits and deletions in this context).

5.2 Morpheme Boundary Detection

We will now highlight some observations based on the segmentation output of the algorithms underscoring some special cases that were handled/not handled by the algorithms.

As shown in Table 6, we notice that Linguistica only separates the final suffix, which in this example is the plural ending; Morfessor shows a tendency to overly segment the words.

In another instance, the word *lekkaparishOdhakarAgalAraru* was not analyzed either by UnDivide or by Linguistica. Morfessor, however, produced the following segmentation:

[6] UnDivide and Linguistica in their default settings ignore singleton word types. It was possible to change two parameters of UnDivide to accept this change, but such a change was not possible with Linguistica 3.

Algorithm	jagaLavADatoDagidaru	aNNatammaMdiriddaru
Linguistica	jagaLavADatoDagida+ru	aNNatammaMdiridda+ru
Morfessor-CatMAP	jagaLa+vADa+toDagida+ru	aNNatammaMdir+idda+ru
UnDivide	jagaLa+vADatoDagidaru	aNNa+tammaMdiriddaru

Table 6: Sample of detected morpheme boundaries for the three algorithms for two words *jagaLavADatoDagidaru* and *aNNatammaMdiriddaru*.

$$lekka/STM + pari/STM + shOdha/STM + ka/SUF + rAga/STM + lAra/STM + ru/SUF$$

Based on the sample and other segmentations, we observe the ability of Morfessor to deal with the complex morphology of Kannada. In particular, the instance of not splitting the compound *aNNatammandiru* (meaning 'brothers') interesting. However, there are also cases of over-segmentation as seen in the last example here, where *lekkaparishOdhaka*, a compound, has been segmented. It appears that a more data-driven analysis leading to a careful choice of the perplexity parameter *b* is necessary to tune the model.

5.3 Algorithm-specific Features

We will now consider some features we observe when analyzing the segmentation output of each of the algorithms.

5.3.1 Linguistica

A closer examination of Linguistica output reveals that it is particularly weak at segmenting Kannada compound words and its complex verbal inflectional system. Kannada being a highly inflected language with a wide variety of inflectional and derivational morphology acting upon the stems to produce valid words, the single-slot capability of Linguistica is a serious shortcoming of the model. With word formation processes in Kannada governed by compounds, phonotactics (sandhi), prefixation and serial suffixation (from its agglutinative characteristics), the one-slot procedure seems rather simplistic to capture the wide spectrum of word formation processes. Nevertheless, looking at the signatures and the stems taking the signatures, a general framework of a paradigm can be induced and is likely to be more descriptive with a very large data set.

5.3.2 UnDivide Algorithm

- The algorithm was successfully able to generate the following (character-change) rules by a single character replacement, addition and deletion at morpheme boundaries as instantiated below. For the noun stems, *guDisalu, hAvu, hUvu, haDagu, kADu, kAlu, mAtu, nIru, pAlu*

  1. the stem final *u* becomes *i* before the dative case marker *ge*;
  2. the stem final *u* becomes *i* before the genitive case marker *na*; and
  3. the stem final *u* becomes *i* before the dative case marker *niMda*

  which are incipient rules of noun declension (paradigm generation) for Kannada.

- The list of the top suffixes learned by the algorithm includes: *gaLannu, yannu, nnu, vannu, da, lli, nige, gaLu, nannu, ru, ya, diMda, na, ge, yalli, dalli, koMDu*. One can identify these to be the noun endings.
- The algorithm correctly analyzed instances of derivational affixes. As an example, we see that suffixes *shAhi* in *adhikArashAhi* and *kAra* in *itihAsakAra* and *kUlikAra* were identified.
- Although the heuristics to detect composite suffixes seems plausible for Kannada with the successful detection (and segmentation) of composite suffixes *kkiruva* in *pUrvakkiruva* as *kke* and *ruva*, a more reasonable analysis should likely take into consideration the morphophonemic (internal sandhi) rules.

It is also worth pointing out here that despite the claim of the language-independence of the algorithm, the 16 parameters in the algorithm need tuning for a better capturing the morphological nuances of the language under consideration.

5.3.3 Morfessor Categories-MAP

- Upon closer inspection, we noticed that the algorithm has only generated 37 suffixes, which seems very small given the size of the data. However, owing to the small test set, the performance does not reflect this shortcoming. In the larger data, the algorithm generated 196 suffixes.
- The algorithm has the capability to identify prefixes (with its dedicated category for a prefix, apart from stem and suffix). While it correctly identified the prefixes *vi* in *viBinna*, *A* in *AdEsha*, *saM* in *saMpUrNa* and *a* in *amUrta*, it incorrectly segmented *ADuvudu* as *A* - Prefix and *Duvudu* - Stem.
- As in the case of the algorithm UnDivide, a few instances of derivational morphology were successfully analyzed by the algorithm. e.g: The suffix *shAhi* in *adhikArashAhi* and *kAra* in *itihAsakAra* and *kUlikAra* were identified.

Unfortunately owing to the fact that no sufficient segmented data was available to tune the parameters, we are unable to make further language specific generalizations on the items mentioned above. Testing the systems on a large collection of words from real world data is the only way to discover some of the potential problems or interesting segmentation patterns.

6 Analysis of Word Frequency Distributions

We now digress slightly from the unsupervised morphological analysis set up and consider some aspects of Kannada in the light of analyses derived from its word frequency distribution obtained from the EMIL corpus considered in the study. In this context we ask the following questions that we think are important from an IR point of view.

1. We know that in English, one can safely say that the most frequent word in a corpus is ‘the’ and will likely not be very far from truth to say that ‘of’ and ‘and’ follow ‘the’ in the most frequent list of words. Analogously, what words in Kannada are most frequent?
2. We know that Kannada has a richer morphology compared to English. Is it possible to obtain a quantitative comparison of the relative complexities? For instance, how do their vocabulary growths compare?

3. In the realm of IR a stopword list contains nonsignificant words that are removed from a document or a request before beginning the indexing process. What would a list of stopwords for Kannada look like?

The answers to these questions will be the material we explore in this section.

First we consider the word frequency distribution for the word types in the EMILLE corpus (with size about a million word tokens) and obtain the frequency of occurrence of the word types in the corpus. A snapshot of the frequencies of the top ten and the bottom ten are shown in the Table 7.

10 most frequent words		**10 least frequent words**	
10993	mattu	1	beMbalavAgirisikoLLabEkiruvudariMda
8007	oMdu	1	vyavasthApakarAgiruvadillavAddariMda
5374	A	1	AdEshisalpaTTavugaLu
4803	athavA	1	sURyacaMdranakSatragaLiruvavarege
4608	eMdu	1	AtaMkagaLannuMTumADuvavaruivarella
4525	mEle	1	prItivAtsalyavuLLavarAgiruttAre
3692	Adare	1	toMdaregIDAgiruvavareMdare
3388	tanna	1	nayanamanOharavAgirabEkallade
3370	hAgU	1	paThyapustakagaLaMthavirabEkeMbudannu

Table 7: The most frequent and least frequent words in the EMILLE corpus for Kannada.

We notice that the high frequency slots are occupied by conjunctions, articles, number words, postposition and pronouns - words that are right candidates for being included in the stopword list. The words in the other column, the singleton words, may belong to various categories, but what is salient among them is their length. While some are results of typographical errors, others are true words formed from internal sandhi processes or compounding or a combination of both. These singletons account for nearly 66% of the corpus. Successful morphological analysis of these words cannot be achieved by paradigm look-up alone since their stems are outcomes of the productive word-formation processes in Kannada and by their very nature, they cannot all be found in a lexicon. On the other hand, owing to their sparse occurrence in corpora, taken by themselves, they are not amenable to be considered in statistical analyses. Processing them requires a combination of rule-based and stochastic approaches deriving the benefits of linguistic expertise (for rule-based methods) and access to large, diverse and permuted corpora (for stochastic techniques).

Having obtained a frequency list of the word types we observe that the most frequent word, *mattu*, accounts for about 1.3% of all word occurrences, the second most frequent word, *oMdu*, accounts for slightly over 1% of all words, followed by the third which accounts for about 0.8% of all words. Going down the frequency list, we notice that it takes 3458 word types to account for half the corpus (in terms of word tokens).

Now consider these numbers in the light of the following fact that in the Brown Corpus, the word *the* is the most frequently occurring word, and by itself accounts for nearly 7% of all word occurrences (69,970 out of slightly over 1 million). True to Zipf's Law, the second-place word *of* accounts for slightly over 3.5% of words (36,410 occurrences), followed by "and" (28,854).

Only 135 vocabulary items are needed to account for half the Brown Corpus[7].

Using the two facts, we make the following important observations towards making a guesstimate of the size of a stopword list for Kannada.

- Comparing the number of word types required to cover 50% of the corpus (3458 types in Kannada vs. 135 in English) and noting that the corpora are roughly of the same size (and assuming that the genres are similar), we notice that the number of word types for Kannada is about 25 times that for English. Thus we need to look farther than the number of stopwords in English to generate a reasonable list of stop words.
- Shifting our attention to the other end of the frequency distribution, we now consider the growth of vocabulary comparing the proportion of *hapax legomena* or the number of words occurring once in the corpus. Baayen ((Baayen, 2001), pp. 49-50) shows how the growth rate of the vocabulary (the rate at which the vocabulary size increases as sample size increases), can be estimated as, the ratio of the number of *hapax legomena* to the number of tokens. Intuitively, this means that the proportion of hapax legomena encountered up to the N th token is a reasonable estimate of how likely it is that word $N + 1$ will be a hapax legomenon, i.e., a word that we have not seen before and one that will consequently increase vocabulary size.

  For the Brown corpus, the number of *hapax legomena* is 24375 and given its corpus size to be 1015945, we have an estimate of vocabulary growth rate for the Brown corpus is $24375/1015945 = 0.024$. Now consider a similarly diverse corpus, EMIL with the number of *hapax legomena* $= 140353$ and a corpus size $= 997012$. An estimate of the vocabulary growth rate for EMILLE(Kannada) is then $140353/997012 = 0.14$. Discounting foreign words (words with consonant endings are good candidates) totaling to about 4585 words, the estimate of vocabulary growth rate comes to 0.136. Thus, even at a corpus size of nearly 1 million, we notice that the growth of vocabulary for Kannada is roughly about 6 times that of English. This could be construed as an approximate quantitative comparison of the relative complexities in vocabulary between English and Kannada and one could attribute this difference to the rich and complex morphology of Kannada compared to English.

7 Conclusion

In this study we have seen that a set of unsupervised methods of morpheme induction perform morpheme boundary segmentation reasonably well. When trained on a corpus of size of about 990K words, the best performing algorithm (Morfessor) had an F-score of 73%. A key feature of these methods is that they have no language specific rules - the heuristics are language independent and probabilistic relying only on the training corpus, but are nonetheless able to capture some important features of the morphology of Kannada. However, we also saw that for a reasonable coverage of the productive morphological processes, we would need an approach that captures the productive process. This is possible by a synergistic approach to morphological analysis that combines a linguistically grounded, rule-based approach with a stochastic method.

So, where do we go from here? In the comprehensive survey article on unsupervised learning of morphology (ULM)(Hammarström and Borin, 2011), the authors, summarizing the general

[7] Source: http://www.edict.biz/textanalyser/wordlists.htm

strengths and weaknesses of the methods, state “typically word segmentation algorithms perform on an insufficient level, apparently due to the lack of any notion of morphotactics. On the other hand, typical morphology learning algorithms have problems because the ingrained assumptions they make about word structure are generally wrong (i.e., too strict) for Finnish or for other highly inflecting or compounding languages. In short, they cannot handle the possibly high number of morphemes per word.”

Continuing their analysis on whether ULM is of any use, “most ULM approaches reported in the literature are small proof-of-concept experiments, which generally founder on the lack of evaluation data. The MorphoChallenge series does provide adequate gold-standard evaluation data for Finnish, English, German, Arabic, and Turkish as well as task-based Information Retrieval (IR) evaluation data for English, German, and Finnish. It can be seen that ULM systems are mature enough to enhance IR, but so far, ULM systems are not close to full accuracy on the gold standard.”

So if our immediate goal is stemming for IR in Kannada, there is some hope in the pursuit of a hybrid (unsupervised and rule-based) stemmer for Kannada utilizing the key ideas of the algorithms considered in this study. Standing on the shoulder of giants such as English and Finnish (and possibly Turkish) we should look for models attempted and benefit from the knowledge advances achieved for making progress in the task of morphological segmentation of words and eventual stemming in Kannada.

Acknowledgements

We would like to acknowledge the help of the members of Sriranga Digital Software Technologies - Prof. C. S. Yogananda and D. Shivashankar, who not only made the corpus of a collection of short stories in Kannada available for this study but also made it computation friendly with the use of their unicode to ASCII converter.

References

The emille/ciil corpus, catalogue reference: Elra-w0037.

Antony, P., Kumar, M., and Soman, K. (2010). Paradigm based morphological analyzer for kannada language using machine learning approach. *International Journal on Advances in Computational Sciences and Technology ISSN*, pages 0973–6107.

Baayen, R. (2001). *Word frequency distributions*, volume 18. Springer.

Creutz, M. and Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3.

Dasgupta, S. and Ng, V. (2006). Unsupervised morphological parsing of bengali. *Language Resources and Evaluation*, 40(3):311–330.

Dasgupta, S. and Ng, V. (2007). High-performance, language-independent morphological segmentation. In *Proceedings of NAACL HLT*, pages 155–163.

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

Hammarström, H. and Borin, L. (2011). Unsupervised learning of morphology. *Comput. Linguist.*, 37(2):309–350.

Keshava, S. and Pitler, E. (2006). A simpler, intuitive approach to morpheme induction. In *Proceedings of 2nd Pascal Challenges Workshop*, pages 31–35.

Larkey, L. S. and Connell, M. E. (2003). Structured queries, language modeling, and relevance modeling in cross-language information retrieval. In *Information Processing and Management Special Issue on Cross Language Information Retrieval*.

Murthy, K. (1999). A network and process model for morphological analysis/generation. In *ICOSAL-2, the Second International Conference on South Asian Languages, Punjabi University, Patiala, India*.

Ramasamy, V., Antony, P., Saravanan, S., and Soman, K. (2011). A rule based kannada morphological analyzer and generator using finite state transducer. *International Journal of Computer Applications*, 27(10):45–52.

Shastri, G. (2011). Kannada morphological analyser and generator using trie. *IJCSNS*, 11(1):112.

Sridhar, S. (1990). *Kannada*. Routledge.

Vikram, T. and Urs, S. (2007). Development of prototype morphological analyzer for the south indian language of kannada. asian digital libraries. looking back 10 years and forging new frontiers. *Lecture Notes in Computer Science Springer*.