Morphological Cluster Induction of Bantu Words Using a Weighted Similarity Measure

Catherine Chavula
University of Cape Town
Private Bag X3
Cape Town, South Africa 7701
cchavula@cs.uct.ac.za

Hussein Suleman
University of Cape Town
Private Bag X3
Cape Town, South Africa 7701
hussein@cs.uct.ac.za

ABSTRACT

Unsupervised morphological segmentation is attractive to low density languages with little linguistic description, such as many Bantu languages. However, techniques that cluster morphologically related words use string similarity metrics that are more suited to languages with simple morphological systems. The paper proposes a weighted similarity measure that uses an approach for calculating Ordered Weighted Aggregator (OWA) operator weights based on normal distribution. The weighting favours shared character sequences with high likelihood of being part of stems for highly agglutinative languages. The approach is evaluated on text for Chichewa and Citumbuka, which belong to the group N of Guthrie Bantu languages classification. Cluster analysis results show that the proposed weighted word similarity metric produces better clusters than Dice Coefficient. Morpheme segmentation results on clusters generated using the OWA weights metric are comparable to the state-of-the-art morphological analysis tools.

CCS CONCEPTS

•Computing methodologies → Phonology / morphology;

KEYWORDS

unsupervised morphological segmentation, similarity metrics

ACM Reference format:
Catherine Chavula and Hussein Suleman. 2017. Morphological Cluster Induction of Bantu Words Using a Weighted Similarity Measure. In *Proceedings of SAICSIT '17, Thaba Nchu, South Africa, September 26–28, 2017,* 9 pages.
DOI: 10.1145/3129416.3129453

1 INTRODUCTION

Morphological analysis is the task of breaking words into their constituent morphemes and labelling them with their functional classes. A typical morphological parsing system would produce *cat+N+PL* for an input *cats*. Morphological analysis is known to improve the performance of tasks in Natural Language Processing (NLP) and Information Retrieval (IR). However, this is challenging for low density languages with complex morphological systems, such as Bantu languages. Techniques for morphological analysis that have produced significant results for agglutinative languages use resources such as lexicons and hand-coded grammar based rules [12].

Fortunately, with the current advances of Machine Learning (ML) techniques, unsupervised techniques are able to learn the morphological structure of words from raw text. Current approaches focus on morpheme segmentation [5], the process of dividing words into constituent morphemes, i.e., stem and affixes, without specifying their functional labels. For example, a morpheme segmentation tool would produce *cat+s* for an input *cats*. This is beneficial to languages such as Bantu languages because tools can be bootstrapped without detailed linguistic information of the language. Additionally, unsupervised methods eliminate the need for linguistic resources such as the lexicon [12].

Bantu languages belong to a family of languages consisting of over 400 languages spoken in Sub-Saharan Africa. Bantu languages were grouped into several classes using an alphanumeric system by Guthrie [11]. For example, Chichewa and Citumbuka are in zone N and have labels N30 and N21 respectively [6, 15]. Bantu languages are among the low density languages of the world. Bantu languages are agglutinative and have similar morphological systems [19]. Bantu word structure consists of both prefixal and suffixal morphological slots. This is true for Bantu languages that use conjunctive writing system, i.e., a system where morphological elements are orthographically realised connected or combined as a single word [19].

Some of the techniques that are used in unsupervised morphological learning rely on grouping words, which are likely morphological variants to learn the morphology of the language [12]. Such methods use word similarity metrics to compare if two words are orthographically similar. However, current string similarity metrics do not fairly accommodate the distribution of morphemes for morphologically complex words of languages such as Bantu languages. This paper proposes a weighted metric that uses an Ordered Weighted Aggregator (OWA) operator to cluster morphologically related words based on shared character patterns. The proposed OWA metric is calculated based on normal distribution and favours shared patterns of characters occurring towards the centre of words. The assumption is that Bantu stems are morphemes that occur inside a word (dependent on word and sequence length) and not on the borders of the word. Clustering algorithms such as Hierarchical

SAICSIT '17, Thaba Nchu, South Africa

978-1-4503-5250-5/17/09…$15.00
DOI: 10.1145/3129416.3129453

Agglomerative Clustering (HAC) are used with a morpheme segmentation algorithm that uses heuristics tailored to Bantu words *prefix + stem + suffix* to come up with linguistically motivated roots.

The remainder of the paper is structured as follows. Section 2 discusses related works. Section 3 describes the morphological structure of Bantu Languages. Section 4 proposes the pipeline for morphological segmentation and the OWA metric for morphological cluster induction. Section 5 describes the development of data sets, experiments and results. Section 6 discusses the results. Section 7 concludes and provides future work.

2 RELATED WORK

Several approaches for morpheme segmentation have been proposed in literature. Unsupervised morpheme segmentation techniques that group words into morphological paradigms have used statistical [3, 16], semantic [22, 25] and syntactic [5] methods. These methods use a clustering algorithm or approach and a word similarity metric to come up with morphological clusters.

2.1 Unsupervised Morphological Paradigm Induction

Unsupervised clustering algorithms such as HAC algorithm have been used to come up with word clusters [1, 3, 16]. However, different similarity metrics have been used. Adamson and Boreham [1] use Dice's coefficient to compute string similarity based on shared bi-grams on words. Using generic string similarity measures does not work well with highly inflectional languages with prefixation and suffixation. Majumder et al. [16] proposed a new metric for string similarity, which was applied on French and Bengali text to cluster words using HAC algorithm. Their approach is similar to the work proposed in this paper because they use orthographic similarity to compare words as well as HAC. However, their similarity metric is tuned to identify suffixes. Bhat [3] used the metric proposed by Majumder et al. [16] on Kannada text to create morphological classes for a statistical stemmer. The results obtained on these studies were comparable. Semantic methods are applied on raw text to come up with semantic groups. These approaches use local context to come up with words that appear in the same context. The assumption is that words that appear in the same context have the same meaning. Latent Semantic Analysis (LSA) has been used to come up with semantic classes from raw text [22]. Recent advances in machine learning have made semantic analysis easier. Üstün and Can used word2vec word embeddings to come up with semantic vectors from raw text [25]. Cosine similarity was used to check orthographic similarity of words within the same group. The approach is compared to the Morfessor baseline family. The Morfessor family uses probabilistic Maximum a Posteriori (MAP) models to choose the optimum model criteria [7].

2.2 Morphological Processing for Bantu Languages

Different approaches to morphological analysis have been applied on Bantu languages. Munro and Manning investigated whether language specific and language independent morphological analysis techniques improve classification of Chichewa short text messages [18]. The language independent word analyser was adapted from Goldwater et al. [9] and uses Hierarchical Dirichlet Process (HDP). The study concluded that both approaches improve precision of a short message classifier and that the tools have comparable performance. Nguni languages morphological structure similarities have been investigated in the context of two level finite state morphological transducers [4, 20]. The results in these studies have been promising and development of morphological analysers of multiple languages were bootstrapped based on a morphological analyser of a single language. Unsupervised morphological analysis has also been applied on several Bantu languages. Spiegler et al. [24] investigated supervised, semi-supervised and unsupervised morphological segmentation methods on IsiZulu. The supervised method had the best performance followed by semi-supervised and unsupervised. Pauw and Waiganjo [8] investigated morphological analysis using unsupervised maximum entropy learning and the performance of the approach was promising.

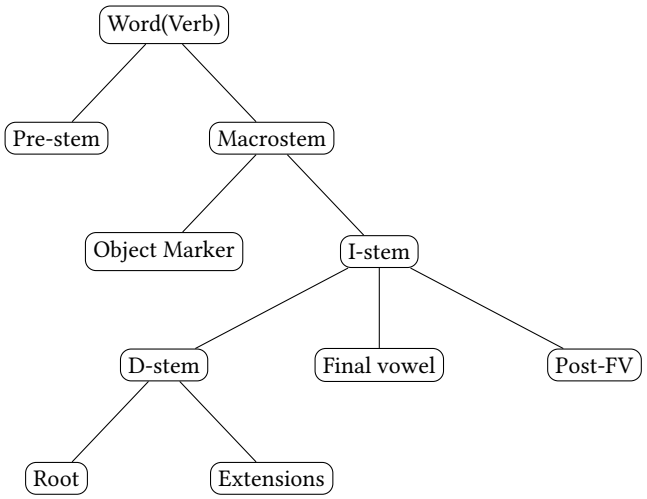

Figure 1: Example of Hierarchical structure of Bantu Verb [19]

3 MORPHOLOGICAL STRUCTURE FOR BANTU LANGUAGES

Bantu languages are morphologically agglutinating languages – inflectional and derivational morphology and other word formation processes are done by concatenating morphemes onto root or existing words together. Reduplication and compounding are very common word formation processes. Accordingly, Bantu words have a complex morphological structure, which is highly evident in verbs. Verbal inflection expresses features that are lexically expressed in other languages. For example, in Chichewa, the verb *ti+dza+mu+fun+a+be* which means *we will still want her/him* expresses *subject ,tense and aspect, and object* in one word. Interestingly, Bantu languages, especially closely related languages, exhibit a lot of similarities in terms of morphology for all word categories. Linguists have proposed a Proto-Bantu morphological structure of verbs, which attempts to account for the morphological system of Bantu verbs [19]. Figure 1 shows the hierarchical structure proposed for the Proto-Bantu verbs [19]. Pre-stem consists of in-

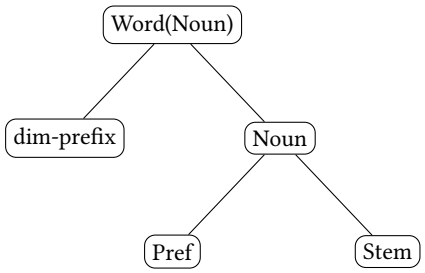

Figure 2: Example structure of a noun [14]

flectional morphemes that combine with the macro-stem to form a verbal word. The root and extension combination is called the derivational stem. In some cases, the root or derivational stem may be reduplicated to form complex words. The derivational stem, when combined with the final vowel, is called an inflectional stem. This structure has been schematised as a slot system, in which some slots are mutually exclusive. Each slot can consist of zero or more morphemes. The slot schema found in the Chichewa language is given by Mchombo [17] as follows:

NEG + SM + TA + MOD + OM + ROOT + [EXT] + FV

These slots indicate morphemes for the functions such as negation (NEG), subject marker (SM), Tense and Aspect (TA), Mode (MOD), object marker (OM) and verb extension or derivation suffixes (EXT) [17].

Similarly, other word categories, such as adjectives and nouns, may consist of multiple morphemes. For example, noun inflection is marked by the class prefix – nouns are classified into classes. Moreover, multiple affixes marking a class can be added to a word, e.g., the diminutive class affixes can be added to a noun with its own class prefix. Due to the agglutinative nature of Bantu languages, nouns for languages such as Chichewa combine with demonstratives in a suffixation process [17]. For example, the phrase these *these cups* in English is *ti+ma+kapu iti* in Chichewa but usually a short form also known as enclitic is used – *ti+ma+kapu+ti*. Figure 2 shows a simple structure of the noun [14, 17].

Therefore, Bantu words consist of multiple morphemes with different functions. In order to cluster words that are morphological variants using character level word analysis, character sequences should not be treated equally: character sequences that are likely prefixes and suffixes should be penalised. The proposed weighted metric assigns higher weights to sequences found inside a word when comparing words for orthographic similarity. Additionally, the morpheme segmentation algorithm in section 4 uses this knowledge to estimate morpheme boundaries.

4 MORPHOLOGICAL CLUSTER INDUCTION AND SEGMENTATION

Unsupervised morphological learning approaches that cluster morphological variants use different features to group words together. The paper reports on a study that uses shared character sequences as measure of similarity to group words together. As shown in Figure 3, the proposed morpheme segmentation approach is divided into four independent steps: (1) words with similar surface forms are grouped into clusters based on the OWA string distance similarity measure described in Section 4.1; (2) an algorithm is used to determine the boundaries between the prefix, stem and suffix; (3) an algorithm tailored for Bantu words is used to learn affixes; (4) the words are divided into their constituent morphemes.

4.1 Normal Distribution Based OWA Weights

The proposed general structural view of any word consists of the prefix, stem and suffix. Using sequences of patterns such as n-grams as features for word similarity, the stem is always part of the internal morphemes. String similarity metrics using n-grams as features for similarity such as dice coefficient, do not consider the position of the patterns in the compared strings, which may lead to high similarity values for words with similar affixes. The proposed weighted measure assigns higher weights to character sequences likely to be roots with the assumption that roots provides the core meaning of a word. 3-grams are used because the common Bantu root structure is *-CVC-* { *-fun-*} for verbs. The proposed Normal distribution based OWA Weights are assigned to similar character sequences based on the position of the pattern in a word.

An Ordered Weighted Aggregation (OWA) Operator is a collection of operators proposed by Yager [27] for aggregating information usually from multiple conflicting criteria. An OWA operator of dimension n is a mapping, OWA:$R_n \rightarrow R$, with an n vector w = $(w_1, w_2, ..., w_n)^T$ such that $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.
A collection of n aggregated arguments or objects $a_1, a_2, .., a_n$ takes a form of n preferences provided by n different individuals , criteria or objects. The OWA averaging is performed as follows:

$$OWA_w(a_1, a_2, ..., a_n) = \sum_{j=1}^{n} w_j b_j \quad (1)$$

where b_j is the jth largest element of the collection of the aggregate objects. The primary challenge to OWA aggregation is to determine the weights [27]. Xu [26] proposed a normal distribution-based method for calculating OWA weights. A normal distribution based OWA weighting assigns low weights to preferences or values away from the central value. This is analogous to calculating the similarity metrics of two words where the stem or morphemes are highly likely to be internal morphemes. Therefore, a normal distribution-based OWA can be applied to determine similarity of character patterns found in any two terms t_1 and t_2. Both t_1 and t_2 can be divided into 3-grams to generate vectors of length $|t_1|$-2 and $|t_2|$-2. The longer vector is used to specify the value of n which is the dimension of the OWA vector. The weights in the OWA vector are used to calculate the overall similarity metric. $a_1, a_2, .., a_n$ objects corresponds to 3-gram vector of the longest string. The value b_i is generated from a similarity vector consisting of 0's and 1's for positions with no matching patterns and similar patterns respectively.

The weights for character sequence patterns based on 3-grams is estimated using Xu's [26] OWA weights normal distribution based formulation as follows:

$$w_i = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-[(i-\mu_n)^2/2\sigma^2 n]} i = 1, 2, ..., n \quad (2)$$

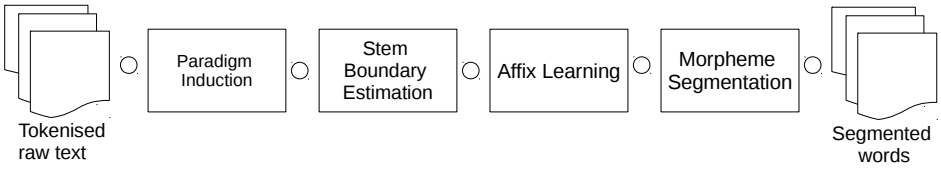

Figure 3: The proposed approach has four parts: (1) clustering the raw tokenised text into morphological groups; (2) determining the stem of a word to compute the word boundary; (3) learning affixes; and (3) segmenting words into their constituent morphemes

$$\mu_n = \frac{1+n}{2} \tag{3}$$

$$\sigma_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(i-\mu_n)^2} \tag{4}$$

The proposed weighted similarity measure is based on weighting 3-grams by their position in a word using a method for estimating weights for an OWA operator. The mean and variance of the used normal distribution depends on the length of words being compared, e.g., the mean is the centre of the word given by the position of the middle 3-gram. However, word similarity measures such as Dice coefficient treat all sequences equally regardless of their position.

Dice coefficient word similarity metric can be easily calculated when character sequences are used to compare words. The Dice coefficient of two terms t_1 and t_2 can be calculated as follows when using 3-grams:

$$\text{Dice coefficient} = \frac{2c}{(|t_1| + |t_2|) - 4} \tag{5}$$

where c is the number of common 3-grams.

To illustrate the difference between the proposed weighted similarity measure and Dice coefficient, the following provides examples of calculating the metrics for words of different lengths and roots. Strings of length less than three are not interesting as the patterns of strings being considered are 3-grams. Comparing two strings both of length 3 yields to zero unless the words are the same. Words with four characters in the proposed method receive the same treatment as in Dice Coefficient, i.e. two 3-grams are generated and each gets a weight value of 0.5. For example, suppose three words, *funa (fun , una)*, *fika (fik , ika)* and *zika (zik , ika)*, v_1, v_2 and v_3 respectively. The computations for their similarity measures is as follows:

Using owa weights:

v_1 and v_2 = $0 * 0.5 + 0 * 0.5 = 0$

v_2 and v_3 =$0 * 0.5 + 1 * 0.5 = 0.5$

v_1 and v_3 = $0 * 0.5 + 0 * 0.5 = 0$

Using Dice coefficent:

v_1 and v_2 = $\frac{2(0)}{(4+4)-4} = 0$

v_2 and v_3 = $\frac{2(1)}{(4+4)-4} = 0.5$

v_1 and v_3 = $\frac{2(0)}{(4+4)-4} = 0$

Comparing strings of length 5 shows how the proposed method treats patterns of characters of three positions. The weights for the three positions are 0.2429, 0.5142 and 0.2429 respectively. The outer weights are smaller in size as these positions are assumed be that of affixes and therefore not interesting. Assume three words, *afika (afi , fik, ika)*, *afike (afi, fik , ike)* and *zika (azi, zik , ika)*, v_1, v_2 and v_3 respectively. Their similarity values using the two metrics is as follows:
Using owa weights:

v_1 and v_2 = $0.2429 * 1 + 0.5142 * 1 + 0.2429 * 0 = 0.7571$

v_2 and v_3 = $0.2429 * 0 + 0.5142 * 0 + 0.2429 * 0 = 0$

v_1 and v_3 = $0.2429 * 0 + 0.5142 * 0 + 0.2429 * 1 = 0.2429$

Using Dice coefficent:

v_1 and v_2 = $\frac{2(2)}{(5+5)-4} = 0.6667$

v_2 and v_3 = $\frac{2(0)}{(5+5)-4} = 0$

v_1 and v_3 = $\frac{2(1)}{(5+5)-4} = 0.3333$

v_1 and v_2 are very similar because they share the same root and therefore, should be in the same morphological cluster. v_1 and

v_3 are only similar in terms of affixes and their similarity metric should be very low. The proposed method assigned a higher word similarity score to the morphological variants than that assigned by Dice Coefficient. Similarly, words with similar affixes scored lower than that of Dice Coefficient.

Table 1 shows an example for calculating similarity metrics using the normal distribution based weights for words *tikufunika*, *tikufika* and *timafunika*, 3-gram vectors represented by v_1,v_2 and v_3 respectively. The aggregated weight is the sum of weights where there are matching 3-grams.

Table 1: An example of calculating similarity metrics using a normal distribution based OWA operator for Chichewa words: *tikufunika* , *tikufika* and *timafunika*

i	1	2	3	4	5	6	7	8
w_i	0.0588	0.1042	0.1525	0.1845	0.1845	0.1525	0.10427	0.0588
v_1	tik	iku	kuf	ufu	fun	uni	nik	ika
v_2	tik	iku	kuf	ufi	fik			ika
v_3	tim	ima	maf	afu	fun	uni	nik	ika

Similarity metrics using Dice coefficient and OWA weighted metrics for v_1,v_2 and v_3:

Dice coefficient are as follows:

v_1 and v_2 = $\frac{2(4)}{10+8-4}$ = $\frac{8}{14}$ = 0.571428

v_1 and v_3 = $\frac{2(4)}{10+10-4}$ = $\frac{8}{16}$ = 0.5

v_2 and v_3 = $\frac{2(1)}{10+8-4}$ = $\frac{4}{14}$ = 0.142857

OWA weights:

v_1 and v_2 = 0.0588 + 0.1042 + 0.1525 + 0.0588 = 0.3743

v_1 and v_3 = 0.1845 + 0.1525 + 0.10427 + 0.0588 = 0.5007

v_2 and v_3 = 0.0588

Although (v_1 and v_2) and (v_1 and v_3) have the same similar number of shared patterns of 3-grams, the OWA operator metric has a slightly higher value for the words that have a common stem or morphological variants. Therefore, the OWA operator reduces the effect of similar character sequences that are on the border of a word.

4.2 Cluster Induction

Grouping words into morphologically related clusters is done twice to obtain clusters with high purity. Purity is a measure of how similar members of the cluster are. Two algorithms are used in the clustering step: a simple algorithm given in algorithm 1, to create initial clusters and an HAC algorithm. However, HAC is resource hungry and requires ad hoc selection of parameters and thresholds [12, 16]. The first clustering groups words into initial clusters and HAC is used to cluster words within the pre-computed smaller clusters.

Algorithm 1 Morphological Cluster Induction

```
function Cluster Induction(threshold)
    Generate 3-grams for each word w in corpus C
    Create clusters                          ▷ stores seen words
    for each given word w_i in C do
        if w_i already exists in clusters then
            continue
        else
            Create new cluster cluster
            append w_i to the current cluster cluster
            append w_i to clusters
        end if
        for each word w_j in C do
            if i == j then
                continue
            end if
            Find the longest string long between w_i and w_j
            Find the common n-grams com for w_i and w_j
            Compute owa measure for com based on long
            if owa > threshold then
                if w_j already exists in clusters then
                    continue
                else
                    append w_j to the current cluster cluster
                    append w_j to clusters
                end if
            end if
        end for
    end for
end function
```

Clustering is considered as unsupervised learning since items are assigned to a class without prior knowledge of membership. Hierarchical clustering is achieved using two approaches, namely, top-down and bottom-up [13] . Bottom-up clustering is called Hierarchical Agglomerative Clustering (HAC). HAC starts with clusters of a single member and similar clusters are merged recursively until all the clusters are merged to a single cluster. The pair of clusters to merge is identified by a linkage method based on a dissimilarity measure or distance. Common linkage methods include Complete-linkage, average-linkage, wards method and a single link [13]. Complete-linkage considers maximum inter-group distance among pairs of clusters and merges pairs with the smallest maximum distance. Average–linkage merges clusters with smallest average inter-group distance. Single–link links clusters pairs with two most similar items. Ward linkage merges groups with the least variance.

The average-linkage method was used as a merging method by HAC after experimenting with several other methods. Additionally, Majumder et al. conducted analysis on Hindi data and found that the average-linkage was a better method for this task [16]. To choose the number of clusters, cophenetic correlation coefficient, i.e., a cluster quality measure, for a range of thresholds (2 to 4) was calculated [21, 23]. The threshold with the best cophenetic correlation coefficient was chosen. However, number of clusters can be used to select the threshold to be used. This was used in the

analysis in Section 5 as the number of clusters was already set by the reference data set.

4.3 Morpheme Segmentation and Affix Identification

Morphological clusters are used to learn the affixes found in the corpus. Initially, the root of the words in a given cluster is determined by finding the common sub-strings for all word pairs in the cluster. The longest most frequent sub-string is chosen as a likely root in the cluster. An algorithm tailored for Bantu words is used to find affixes and morpheme boundaries of all words in the cluster. The chosen root is used to divide a word into three segments: prefix, stem and suffix.

An algorithm that learns the affixes in both the prefix and suffix segments is used to segment each word into its possible morphemes. The algorithm uses the following basic knowledge for Bantu words: (1) prefixes are open syllable *(V,CV,CCV,CCCV,CCCCV)*; and (2) morphemes in the suffix segment are divided from the last vowel to the first character, usually removing the last vowel when it is either 'a' or 'e'. The affixes learnt from all the words are used to generate possible affixes of the language using the Greenberg Principle [10, 12]. The Greenberg Principle stipulates that if a character sequence appear after or before at least two roots or stems then it is likely to be an affix. For example, a square is given by four words with w_1 , w_2 , w_3 and w_4 with the form stem_i+suffix_x, stem_i+suffix_y, stem_j+suffix_x and stem_j+suffix_y. This also corresponds to prefixes. Any morpheme arrangement found in the suffix or prefix segments were selected for affix generation.

The affixes were derived based on the possible morpheme slots as discussed in Section 3. Each word is assumed to have nine slots regardless of its part of speech. The prefixal component has five slots and the suffixal segment has three slots (final vowel, verb extension and extra suffixal morphemes), and the sixth slot is for the root. Additionally, each slot is assumed to be for a monosyllabic morpheme. This approach provides interesting results since it is closer to function labelling of morphemes, which is a major limitation for unsupervised methods. The process for affix identification is divided into the following steps:

(1) Determine the root of the words in a given cluster
(2) Divide every word in a cluster into three segments: prefix, root and suffix.
(3) Create Greenberg square for the prefix and suffix segment. Keep segments that are only in these squares.
(4) Generate a list of prefixal morphemes by dividing the prefix segment on syllables from the leftmost end.
(5) Generate a list of morphemes by splitting the suffix segment from the rightmost end using length and type of end characters. For example, if the segment is just one character long and it is a vowel, add it as a final vowel.

5 EXPERIMENTAL RESULTS

This section describes the data set used and the experiments conducted to evaluate the proposed approach and the results obtained in the experiments.

5.1 Data Set

The data used in the experiments are made up of two Bantu languages: Chichewa and Citumbuka, as spoken in Malawi. This data was obtained from a raw corpus from Fuko newspaper published by Nations Publications Malawi. Fuko Newspaper is a local newspaper published fortnightly in Chichewa and Citumbuka. 10 volumes of both languages were used to prepare a dataset for the experiments. Firstly, text content was extracted from the newspaper. This data was cleaned to remove punctuation marks and any bad characters and numbers. Then, the data was tokenised and, 25, 210 words for Chichewa and 24,101 words in Citumbuka were realised. The data was checked to remove any illegal words such as English words, named entities, and repeated words. In total 4,813 Chichewa tokens and 4,244 Citumbuka types were obtained. The words in the data set were manually segmented by three linguistics students who are also native speakers of the languages. The words were also manually grouped into morphological clusters. Clusters with single words were removed and 474 clusters for Chichewa and 442 clusters for Citumbuka were found. The reference data for clusters was slightly different from the data set used in other parts of the study, i.e., this was used only in the clustering evaluation task only and had fewer words (single word clusters were thrown away).

Language	Number of types	Clusters
Chichewa	4,813	474
Citumbuka	4,244	442

Table 2: Summary statistics for the data set

5.2 Cluster Analysis and Evaluation

The metric was evaluated in terms of the quality of the clusters that it could create and it is compared against Dice coefficient which is the simple method that fits well with the n-gram based similarity metric. Clustering use patterns in the data to group items together and in many cases the number of clusters are not known in advance. HAC deletes or merges together clusters above a threshold in order to obtain a certain number of clusters [16]. Therefore, thresholds affect the quality of the clusters: the bigger the number of the threshold, the less the number of clusters . Figure 4 shows the relationship between the number of clusters and the threshold for OWA and Dice Coefficient based clusters for Chichewa and Citumbuka respectively. It is possible to empirically determine the number of clusters by plotting the number of clusters produced by a range of thresholds to obtain a region where the number of clusters tend to stabilise. Unfortunately, the data set used was very small and this effect was not seen in the data. The evaluation used a threshold that produced clusters closer to the number of clusters created by human assessors.

5.2.1 Cluster Evaluation. The quality of the clusters was evaluated using purity metric. Purity is an external metric that calculates the percent of objects or items that have been correctly classified and takes the values between 0 and 1. Formally, purity is define as

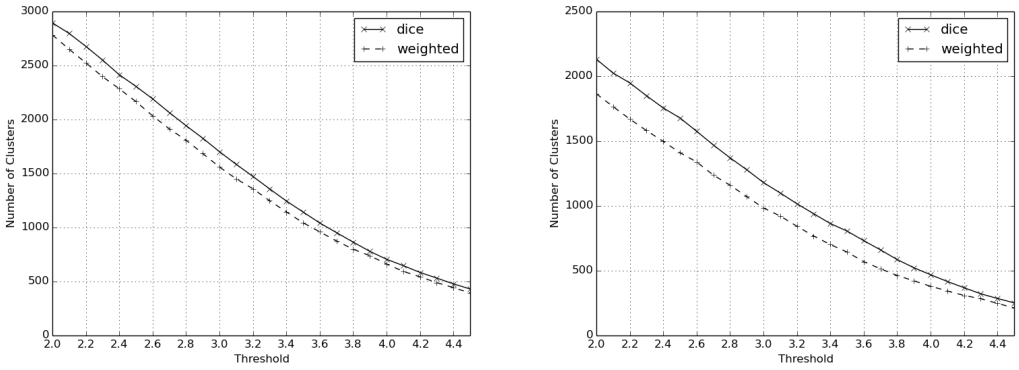

Figure 4: Plots of number of clusters against threshold for using the weighted approach and Dice coefficient for Chichewa and Citumbuka respectively

follows:

$$Purity = \frac{1}{N}\sum_{i=1}^{k} max_j \mid c_i \cap t_j \mid \quad (6)$$

where N = number of items to be classified, k = number of clusters, c_i is a cluster in C, and t_j is the ground truth classification with the highest number of members in c_i. Using the threshold selected in the previous step, purity was calculated for Chichewa and Citumbuka clusters using OWA and Dice coefficient metric. Table 3 gives the purity values obtained for the four clusters. The data shows that

	Citumbuka	Chichewa
OWA	0.7	0.79
Dice	0.59	0.66

Table 3: Cluster Evaluation based on *Purity* using OWA and Dice on Citumbuka and Chichewa data

the clusters created using OWA metric had more correctly classified terms than using Dice coefficient when attempting to create the number of clusters close to those in the reference set. However, it is possible to get the purity of the OWA metric using Dice coefficient but the number of clusters need to be bigger than what was aimed at.

5.3 Word Morpheme Segmentation Evaluation

The clusters created using OWA were used in the task of morpheme segmentation using a method proposed in Section 3. The performance of the morpheme segmentation experiments were compared with results of the language specific analyser [1] based on the morphological rules of the languages [18]. The analyser is available for Chichewa only in PHP and the code was changed to Python to enable the experimentation. An analyser based on Citumbuka rules using the approach by Munro and Manning [18] was also created. Morfessor BaseLine (MBL) was also used as a language independent implementation approach for morpheme segmentation. The data set that was developed in section 5.1 was used to conduct the evaluation. Table 4 shows results for the proposed approach, Morfessor baseline and language specific segmentation tool. To evaluate the segmentation task, precision for segmentation boundary is calculated. Precision is the ratio of the number of correct boundaries found to the total number of morpheme boundaries made. The boundary predictions are judged against the boundaries given in the gold-standard. Table 4 shows the precision values using Morfessor 2, language specific segmentation approach and the proposed approach.

[1] https://github.com/rmunro/chichewa/blob/master/chichewa_seg_php

Table 4: Precision results for segmentation using the proposed approach, language specific segmentation and Morfessor 2 Baseline.

Precision	Chichewa	Citumbuka
Proposed Approach	0.53	0.54
Morfessor 2 BaseLine	0.58	0.50
Language Specific	0.50	0.74

5.4 Affix Learning Evaluation

Several correct affixes were obtained from the analysis of the prefixal and suffixal segments using the proposed algorithm. Each affix was generated based on the slot it can occupy in a word. Table 5 shows some of the generated correct morphemes and their proposed slots. Only three slots were identified for the prefixal segment instead of five. This may be due to the fact that long prefixal segments are scarce and could not make it into the squares. Out of 210 affixes that were identified for Chichewa data set only 72 were incorrect.

6 DISCUSSION

The proposed word similarity measure produced promising results. Similarity scores for morphological variants are higher than words with similar character patterns that are affixes. The proposed method reduces the effect of similar affixes by assigning very low scores for similar pattens on the edge of the word. Dice Coefficient and other similar metrics treat each pattern equally and leads to

Table 5: Example of affixes generated from the clusters.

Slot	Chichewa	Citumbuka
1	"zi', 'ndi', 'o', 'li', 'chi'	'mu', 'vi', 'vya', 'gha', 'ku'
2	'ngo', 'chi', 'dza', 'sa', 'ka'	'nga', 'chi', 'pa', 'ma', 'ku'
3	'yi', 'zi', 'wa', 'ndi', 'dzi',	'ji', 'ka', 'ku', 'chi', 'ti'
Final vowel	'a', 'e', 'i', 'o' , 'u'	'a', 'e', 'i', 'o', 'u'
Extension	'ir', 'ik', 'its', 'er', 'ets'	'isk', 'ir', 'ik', 'il', 'esk', 'er'
enclitic	'nso', 'wo', 'chi', 'mo', 'po'	'ko', 'so', 'po'

higher scores for words which share affixes. The evaluation of the weighted metric in the task of morphological cluster induction produced better clusters in terms of purity than clusters based on Dice coefficient. Evaluation was done on clusters closer to the number of clusters in the cluster benchmark created by humans. However, a value based on a single threshold may not provide a lot of insights on the differences in clustering using the two measures. Perhaps using different thresholds may provide more information about the performance of the two measures.

Unsupervised morphological segmentation has not been widely studied for low density languages. Additionally, related language contexts provide opportunities to streamline the tasks in the pipeline of morphology learning tasks because outputs, word and morpheme distribution are similar. In the proposed approach, the same parameters and algorithms were used for both languages and the results obtained are comparable across the languages. For example, the same approach was able to identify potential affixes in both languages, and similar results were obtained for the purity test using OWA weights and Dice Coefficient.

The proposed weighted metric based on OWA produced better clusters in terms of purity than using Dice Coefficient. This may be attributed to the weighted character sequences based on position in the string and length of the word. This ensured that the similarity metric is less affected by character sequences on the boundary. However, deciding on the number of clusters is an ad hoc process which requires a lot of data analysis: the bigger the number of clusters, the smaller and more similar are the clusters.

The segmentation results show that the proposed approach for morpheme segmentation has performance that is comparable to the state of the art tools. Morfessor performed well on compounds and reduplicated words. For example, reduplicated words such as *chomenechomene* and *pachokopachoko* were segmented as *chomene + chomene* and *pa + choko + pa + choko* respectively. The other two approaches produced *pa + chokopachoko* and shows the failure to identify reduplicated words. However, Morfessor was not able to predict morpheme boundaries at micro level, e.g., single character morphemes like final vowel in verbs. For example, words like *vikunangika* and *vikuwoneka* were segmented as *viku + nangika* and *vi + kuwoneka* instead of *vi+ku+nang+ik+a* and *vi+ku+won+ek+a* respectively. The language specific tool produced correct results for these two words. The proposed approach produced *vi+kunang+ik+a* and *vi+kuwon+ek+a* respectively. The Chichewa segmentation approach using grammar rules [17, 18] was not able to handle corpus specific issues such as spelling variations due to orthographic and phonological differences. The Citumbuka tool had additional rules to address some of the issues in the corpus and had considerably better performance. Still, compounding and reduplication were not handled as this may require a lexicon.

The major issues in the proposed morpheme segmentation approach are: (1) to estimate the root in the morphological cluster; and (2) to identify words with similar surface forms but different meaning. For example, a cluster would have words {*madzi/water, a+ma+dziwa+a/they know, ma+dziwa/this water, madziwo/that water*}. These words need to be segmented differently but using character sequences to group words does not learn the morphology of the language or use any language knowledge.

7 CONCLUSION

The paper has proposed OWA as a weighted word similarity measure for clustering morphologically related words and the results obtained are promising. However, the distribution of affixes and stems for some words is skewed [2, 12] and a normal distribution (symmetric) based approach may not always give the best results. In addition, similar stems in a language may not have the same senses. In this respect, a semantic approach that uses distributed representation of words may address the problem. Future work will use a skewed distribution model appropriate for Bantu language morpheme distribution to generate the weights for measuring similarity. This will be integrated with distributional semantics to take care of words that do not share senses but orthographic representations [25]. Further, functional labelling of morphemes based on cross-language similarities for Bantu languages will be investigated.

ACKNOWLEDGMENTS

This research was partially funded by the National Research Foundation of South Africa (Grant numbers: 85470 and 88209) and University of Cape Town. The authors acknowledge that opinions, findings and conclusions or recommendations expressed in this publication are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

REFERENCES

[1] George W. Adamson and Jillian Boreham. 1974. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval* 10, 7-8 (1974), 253–260. http://dblp.uni-trier.de/db/journals/ipm/ipm10.html#AdamsonB74

[2] Michela Bacchin, Nicola Ferro, and Massimo Melucci. 2005. A Probabilistic Model for Stemmer Generation. *Inf. Process. Manage.* 41, 1 (Jan. 2005), 121–137. https://doi.org/10.1016/j.ipm.2004.04.006

[3] Suma Bhat. 2013. Statistical stemming for Kannada. *The 4th Workshop on South and Southeast Asian NLP (WSSANLP), WSSANLP-2013* (2013), 25–33. https://doi.org/doi=10.1.1.401.2223

[4] Sonja E Bosch, Laurette Pretorius, Kholisa Podile, and Axel Fleisch. 2008. Experimental Fast-Tracking of Morphological Analysers for Nguni Languages.. In *LREC*.

[5] Burcu Can and Suresh Manandhar. 2010. *Clustering Morphological Paradigms Using Syntactic Categories.* Springer Berlin Heidelberg, Berlin, Heidelberg, 641–648. https://doi.org/10.1007/978-3-642-15754-7_77

[6] Jean Josephine Chavula. 2016. *Verbal Derivation and Valency in Citumbuka.* Ph.D. Dissertation. Centre for Linguistics, Leiden University.

[7] Mathias Creutz and Krista Lagus. 2007. Unsupervised Models for Morpheme Segmentation and Morphology Learning. *ACM Trans. Speech Lang. Process.* 4, 1, Article 3 (Feb. 2007), 34 pages. https://doi.org/10.1145/1187415.1187418

[8] Guy De Pauw and Peter Waiganjo Wagacha. 2007. Bootstrapping morphological analysis of Gikuyu using unsupervised maximum entropy learning. In *Proceedings of the eighth INTERSPEECH conference.* 1517–1520.

[9] Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition* 112, 1 (2009), 21–54. https://doi.org/10.1016/j.cognition.2009.03.008

[10] Joseph H. Greenberg. 1957. Order of Affixing: a Study in General Linguistics. In *Essays in Linguistics*, Joseph H. Greenberg (Ed.). University of Chicago Press, Chicago, 86–94.

[11] Malcolm Guthrie. 1967. *Comparative Bantu : an introduction to the comparative linguistics and prehistory of the Bantu languages.* Farnborough : Gregg.

[12] Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics* 37, 2 (2011), 309–350.

[13] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *ACM Comput. Surv.* 31, 3 (Sept. 1999), 264–323. https://doi.org/10.1145/331499.331504

[14] Francis X. Katamba. 2003. *Bantu nominal morphology.* Routledge.

[15] Andrea Kiso. 2012. *Tense and aspect in Chichewa, Citumbuka and Cisena: A description and comparison of the tense-aspect systems in three southeastern Bantu languages.* Ph.D. Dissertation. Department of Linguistics, Stockholm University.

[16] Prasenjit Majumder, Mandar Mitra, Swapan K. Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. YASS: Yet Another Suffix Stripper. *ACM Trans. Inf. Syst.* 25, 4, Article 18 (Oct. 2007). https://doi.org/10.1145/1281485.1281489

[17] S. Mchombo. 2004. *The Syntax of Chichewa.* Cambridge University Press. https://books.google.co.za/books?id=SRCFoDp88oUC

[18] Robert Munro and Christopher D. Manning. 2010. Subword Variation in Text Message Classification. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10).* Association for Computational Linguistics, Stroudsburg, PA, USA, 510–518. http://dl.acm.org/citation.cfm?id=1857999.1858074

[19] Derek Nurse. 2008. *Tense and Aspect in Bantu.* Oxford University Press UK.

[20] Laurette Pretorius and Sonja Bosch. 2009. Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology. In *Proceedings of the First Workshop on Language Technologies for African Languages.* Association for Computational Linguistics, 96–103.

[21] Sinan Saraçli, Nurhan Doğan, and İsmet Doğan. 2013. Comparison of hierarchical cluster analysis methods by cophenetic correlation. *Journal of Inequalities and Applications* 2013, 1 (23 Apr 2013), 203. https://doi.org/10.1186/1029-242X-2013-203

[22] Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free Induction of Morphology Using Latent Semantic Analysis. In *Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7 (ConLL '00).* Association for Computational Linguistics, Stroudsburg, PA, USA, 67–72. https://doi.org/10.3115/1117601.1117615

[23] Robert R. Sokal and F. James Rohlf. 1962. The Comparison of Dendrograms by Objective Methods. *Taxon* 11, 2 (1962), 33–40. https://doi.org/10.2307/1217208

[24] S. Spiegler, B. Golenia, K. Shalonova, P. Flach, and R. Tucker. 2008. Learning the morphology of Zulu with different degrees of supervision. In *2008 IEEE Spoken Language Technology Workshop.* 9–12. https://doi.org/10.1109/SLT.2008.4777827

[25] Ahmet Üstün and Burcu Can. 2016. *Unsupervised Morphological Segmentation Using Neural Word Embeddings.* Springer International Publishing, Cham, 43–53. https://doi.org/10.1007/978-3-319-45925-7_4

[26] Zeshui Xu. 2005. An overview of methods for determining OWA weights. *International Journal of Intelligent Systems* 20, 8 (2005), 843–865. https://doi.org/10.1002/int.20097

[27] Ronald R. Yager. 1988. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Trans. Syst. Man Cybern.* 18, 1 (Jan. 1988), 183–190. https://doi.org/10.1109/21.87068