Evaluation of combined bi-directional branching entropy language models for morphological segmentation of isiXhosa

Lulamile Mzamo[1[0000-0002-8867-7416]] Albert Helberg[1[0000-0001-6833-5163]] and Sonja Bosch[2[0000-0002-9800-5971]]

[1] North-West University, Potchefstroom, South Africa
[2] UNISA, Pretoria, South Africa
Lula_mzamo@yahoo.co.uk, Albert.Helberg@nwu.ac.za, Boschse@unisa.ac.za

Abstract. An evaluation of the IsiXhosa Branching Entropy Segmenter (XBES), an unsupervised morphological segmenter for isiXhosa, is presented. The segmenter contributes a combined bi-directional branching entropy language model with an option for modified Kneser-Ney (mKN) smoothing. XBES's boundary identification accuracy of 77.44 ± 0.32% is comparable to the benchmark Morfessor-Baseline's 77.2 ± 0.10%. XBES's f1 score, of 58 ± 0.10%, is significantly better than Morfessor-Baseline's 48.9 ± 0.75%. The study shows that mKN smoothing degrades performance on branching entropy-based segmentation of isiXhosa, and suggests that better segmentation performance could be achieved in the unsupervised morphological segmentation of isiXhosa, given more data.

Keywords: natural language processing, unsupervised machine learning, morphological segmentation, branching entropy, isiXhosa.

1 Introduction

Work on the unsupervised learning of isiXhosa text segmentation, the IsiXhosa Branching Entropy Segmenter (XBES), was presented in [1]. This paper presents the bi-directional branching entropy language model implemented in XBES and evaluates the XBES against more metrics than just accuracy.

Human language resources and applications currently available in South Africa are still limited. According to [2] this can be attributed to the dependence on Human Language Technology (HLT) expert knowledge, scarcity of data resources, lack of market demand for African languages, and how the particular language relates to other more resourced languages. Morphological analysis is one of the basic tools in the natural language processing (NLP) of agglutinating languages such as isiXhosa.

IsiXhosa is one of the South African official languages belonging to the Bantu language family, which are classified as "resource scarce languages". IsiXhosa is the second largest language in South Africa with 9.3 million mother-tongue speakers (17% of

the South African population), second only to isiZulu [3]. Although there has been an increase in the tools for South African languages, this increase is from a low baseline. Hence there is still a need for NLP tools [4].

IsiXhosa is closely related to other Nguni languages such as isiZulu, Siswati and isiNdebele and therefore work done in it could easily be bootstrapped to these languages as has been shown in [5]. Nguni languages account for 45.8% % of the South African mother tongue speaker population.

2 Morphological segmentation for isiXhosa

2.1 Morphological Segmentation

Morphological analysis is the task of splitting one token, a word, into its constituent units [6], e.g. the segmentation of a word into morphemes, and classification thereof. Morphemes are the smallest meaning bearing component of a word [7]. In languages with rich systems of inflection and derivation, morphological analysis is needed in information retrieval, translation, etc.

A differentiation is made by [8] between morphological segmentation, which splits words into constituent morphemes, and morphological analysis, which also classifies the identified morphemes. This differentiation originated in [9]. The task handled in this paper is morphological segmentation.

2.2 Morphological segmentation in isiXhosa

IsiXhosa is an agglutinating and polysynthetic language in that it usually has many morphemes per word [7]. It is also fusional/inflectional because morpheme boundaries are sometimes fused and difficult to distinguish, e.g *ukwanda* (to grow) is linguistically segmented as *u-ku-and-a*. The *w* is a result of a fusion between the *u* and *a* vowels.

IsiXhosa words are composed of a root, prefixes, suffixes and circumfixes that attach to the root. The root is the main meaning carrying constituent of the word. A circumfix is the "simultaneous affixation of a prefix and suffix to a root or a stem to express a single meaning" [7]. An example of a circumfix in isiXhosa is the combination "a…ang.." in isiXhosa negation, e.g. *a-ka-hamb-ang-a (he/she did not go).*

Each of the affixes (i.e. prefixes, suffixes or circumfixes) is made up of one or more morphemes. Morphemes follow one another in an order prescribed for each word type [10]. In isiXhosa, most roots are however bound morphemes, meaning that they never appear independently as words which are independently meaningful [11]. They at least appear as stems, which are word roots suffixed with a termination vowel [10], e.g. *and-a* in *ukwanda*.

2.3 Automated morphological segmentation of isiXhosa

One of the earliest reports on automated morphological segmentation of South African languages is that of [12] on the automatic acquisition of a Directed Acyclic Graph (DAG) to model the two-level rules for morphological analysers and generators. The

algorithm was tested on English adjectives, inflection of isiXhosa noun locatives and Afrikaans noun plurals, with a 100% accuracy for isiXhosa noun locatives inflection.

An existing isiZulu morphological analyser [13] was bootstrapped by [5] to other Nguni languages including isiXhosa. The study reported that 93.30% of the words (181) were analysed.

Work on the development of text resources for ten South African languages was presented by [14], including a morphologically analysed corpus for isiXhosa. That morphological segmentation corpus is used in this study as the test corpus. The corpus is rated at an accuracy of 84.66%.

The most recent work for isiXhosa segmentation is that of [15], which introduced a lemmatiser for isiXhosa and [16] who presented the development of a rule-based noun stemmer for isiXhosa. The isiXhosa lemmatiser was evaluated at an accuracy of 83.19% and the noun stemmer showed an accuracy rate of 91%.

3 Unsupervised morphological segmentation

The last works done for morphological segmentation for isiXhosa reported in [1] uses unsupervised machine learning in the morphological segmentation of isiXhosa. This is attractive because it bypasses the need for expensive linguistic experts or annotation of training data.

3.1 Supervision in Machine Learning

There are three modes of training a machine learning model, i.e. supervised, semi-supervised and unsupervised [6]. In supervised learning, the training data contains solution examples that the model must generalise from. Data in unsupervised training is devoid of such, but only creates a model from raw data. Semi-supervised systems use anything in between, from using limited supervised data with large amounts of unannotated data to unannotated data with rules built into the model.

The segmenter evaluated in this paper, XBES, uses unsupervised learning in the morphological segmentation of isiXhosa.

3.2 Unsupervised morphological segmentation works

The earliest works in unsupervised morphological segmentation used a form of accessor variety, where a morpheme boundary is identifiable by the possible number of letters that may follow a sequence of letters [17, 18]. This evolved to using mutual information [19, 20], and different forms of Branching Entropy [19, 21].

Minimum Description Length (MDL) [22] has seen extensive use in unsupervised morphological segmentation, primarily as a measure of fit of the training data to heuristic models and statistical models [23, 24]. The comparative standard used in this study, Morfessor-Baseline [25], uses MDL and Maximum likelihood estimation.

Clustering and paradigmatic models is another popular approach. This involves clustering related words into a paradigm using a similarity measure, identifying the stem, and considering the rest as sequences of affixes [26, 27]. A paradigm is a grouping

of words according to their form-meaning correspondence [28]. The similarity measures used are Latent Semantic Analysis [29], Dice and Jaccard coefficients [6], Ordered Weighted Aggregator operators [27] and affixality measurements [30]. Word context is also another technique that is used to identify similar words [31, 32].

Non-parametric Bayesian techniques have also shown promise, including Pitman-Yor process based models [33, 34] and adaptor grammars [35]. These use Markov Chain Monte Carlo (MCMC) simulation with Gibbs Sampling [36] for inference. Contrastive Estimation [37, 38] is another non-parametric model that is showing elegance and promising results.

A number of studies have used a combination of the above techniques and measures [16, 32].

3.3 Choice of unsupervised segmenter for benchmarks

To place this work amongst other segmenters, a standard in morphological segmentation was chosen for comparison. The benchmark segmenter had to be publicly available and had to have been used for highly agglutinative languages like isiXhosa. The Morfessor-Baseline segmenter [25] was chosen because it has been used as a benchmark extensively and is freely available.

To establish a minimum performance baseline a random segmenter that randomly decides whether a point in a word is a boundary of a segment or not was implemented.

4 Character level language modelling

To estimate the branching entropies, character level language modelling is required. Instead of using two language models, one for each direction, XBES's implementation uses one model for both directions such that a dictionary entry points to a vector of two values, the right branching and the left branching values. This reduced the memory footprint.

Both the unsmoothed and modified Kneser-Ney [39] smoothed language models were implemented. The language model was also extended to include an option for using all possible n-gram levels in one model instead of having a maximum n-gram level limit, i.e., an infinite-gram.

The calculation of the VBE and NVBEs are done as specified in [1] and [40] and stored in the model.

The algorithms are briefly described below.

4.1 Un-smoothed bi-directional Branching Entropy language model flow

The input to the unsmoothed modelling is a list of one directional (left-to-right) n-gram strings with frequency counts (n-gram, f) or a mapping of n-gram string to frequency counts and the process returns a single Bi-directional Branching Entropy Language

Model (BELM) with branching entropy values for both directions. Fig. 1 shows the process of the modelling.

The sorting allows the process to do a single pass through the n-gram strings frequency counts. This is important when dealing with large frequency counts as in this case.

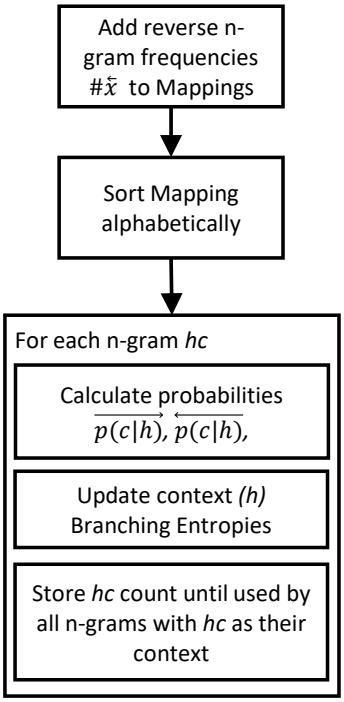

Fig. 1. Un-Smoothed BELM process

The reverse frequencies are updated such that each n-gram *x* is mapped to a list of two counts such that

$$C(x) \rightarrow \begin{pmatrix} \#\vec{x} \\ \#\overleftarrow{x} \end{pmatrix} \quad (1)$$

The branching entropies are calculated according to [41]. The probabilities are discarded after use.

4.2 The modified Kneser-Ney smoothed bi-directional Branch Entropy language model flow

In this exercise, we also wanted to check if smoothing had any effect on the performance of the branching entropy segmenter. Smoothing is necessary where there is data sparsity, which is the case in higher order character n-grams, i.e long n-gram strings.

We implemented a bi-directional mKN smoothed BELM. Input to it is also a list of one directional (left-to-right) n-gram strings with frequency counts (n-gram, f) or a mapping of n-gram string to frequency counts and the process returns a single bi-directional Branching Entropy Language Model (BELM) with branching entropies for both directions. The process is shown in Fig. 2.

The conditional probabilities are calculated according to [39] and the branching entries are according to [41].

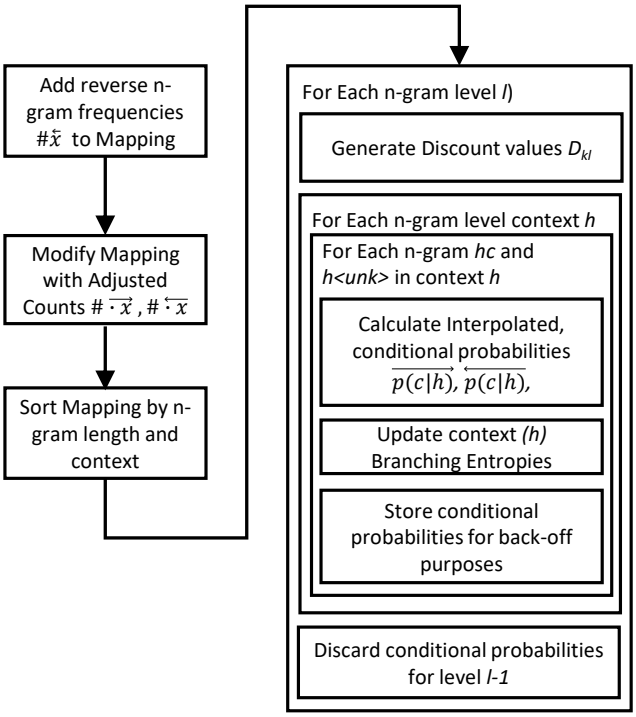

Fig. 2. mKN Smoothed BELM Process

The sorting by n-gram lengths and n-gram contexts ensured that lower level n-grams are processed first as their results are required for the interpolation of higher-level n-grams. This also ensured that discount values could be calculated independently for each level. Lastly, this ensured that n-grams are clustered by context, which is key in mKN smoothing. All this ensured only two passes through the n-gram counts one for

generating discount values and collection interpolation statistics, and one for calculating the probabilities and updating the BELM.

5 Evaluation

This section details the evaluation that was done on XBES.

5.1 Data Sources[1]

A raw unannotated isiXhosa corpus of 1.45 million isiXhosa words was compiled from the isiXhosa version of the South African Constitution [42], isiXhosa text on the internet and the IsiXhosa Genre Classification Corpus [43]. This text is named the training corpus.

For testing purposes the NCHLT IsiXhosa Text Corpus (29 511 tokens) was used.

5.2 Data Splits

For training purposes, ten-fold training was performed for different training set sizes and language model n-gram lengths. The training set sizes chosen were orders of ten (10) from one hundred (100) words to a million words and one and a half million (1.5 million). The n-gram lengths were two (2) to five (5), odd numbers to nineteen (19) and to the maximum n-gram length possible, i.e. the infinite-gram.

For testing purposes a subset of the NCHLT corpus was used. Because the NCHLT corpus was generated with a rule based morphological analyser, the solutions are not all surface segmentations, others include grammatic morphemes. XBES is a surface segmenter and was not built to handle morpheme boundary fusion. As an example the morphological segmentation of *ukwanda* is *u-ku-and-a*. A surface segmenter would segment *ukwanda* to *u-kw-and-a*. Excluding these kinds of entries resulted in an evaluation testing corpus of 13441 tokens.

5.3 Experiment setup

Training was performed for two segmenters, i.e. XBES, and Morfessor-Baseline, using the training corpus, and tested against the testing corpus. The random segmenter does not require training. Both Morfessor-Baseline and XBES were trained with different sizes. Because Morfessor-Baseline does not support specifying n-gram size, only XBES was trained to different model n-gram lengths.

XBES provides an option of using the minimum between the right branching entropy and left branching entropies or the sum of the two. In addition this study tests XBES on unsmoothed and a modified Kneser-Ney smoothed language models [39] as detailed in

[1] The IsiXhosa Genre Classification Corpus and NCHLT IsiXhosa Text Corpus are available at the South African Language Resource Management Agency, (http://rma.nwu.ac.za/index.php)

Algorithms 1 and 2. In addition XBES was evaluated for all the branching entropy modes specified in [1].

Evaluation of the segmentations was measured as boundary identification accuracy and f1 score, where, in a word, a morpheme boundary location is tagged 1 and everything else 0. Accuracy measures how many boundaries and non-boundaries the segmenter identified correctly. The f1 score focuses on the possible boundary location and does not factor the non-boundary word locations.

5.4 Results

The overall results, including the best performance per XBES mode are shown in Table 1. The results are shown with configuration information (i.e. smoothed or not, the operator used to mix the directional branching entropies, the training set size and the language model n-gram level that produced the results).

Table 1. Boundary Identification Results

Method	Highest Accuracy (Smoothing/Op/training Size/n-gram length)	Highest f1 Score (Smoothing/Op/training Size/n-gram length)
Random	50.1 ±0.16	35.7 ±0.16
XBES-BE	71.6 ±0.35 (No/Min/100K/4)	55.3 ±0.12 (No/Sum/1m/7)
XBES-VBE	72.4 ±0.25 (Yes/Min/1.5m/9)	**58 ±0.10 (No/Sum/1.5m/9)**
XBES-N_uVBE	75.8 ± 0.60 (No/Sum/1m/11)	53.6 ±0.35 (No/Sum/1.5m/13-max)
XBES-N_zVBE	**77.4 ±0.32 (No/Sum/1.5m/11)**	55.5 ±1.18 (No/Sum/1.5m/9)
Morfessor-Baseline	**77.2 ±0.10(1m)**	48.9 ±0.75 (10K)

The benchmark ten-fold validation average accuracy from Morfessor-Baseline was measured at 77.2 ±0.10%. The random segmenter presented an average accuracy from ten (10) runs of 50.1 ±0.16%. This implies that any segmenter below this threshold would actively degrades segmentation.

The Random Segmenter's average f1 score was 35.7 ±0.16% whilst Morfessor-Baseline's performance peaked at 10000 words with an average f1 score of 48.9 ±0.75%.

The best 10-fold average accuracy, 77.4 ±0.32%, was achieved by the z-score normalised branching entropy mode (N_zVBE) of XBES at a training set size of one and a half million (1.5 million) words using an unsmoothed 11-gram language model and the *sum* operator. This accuracy, however, is only considered comparable to Morfessor-baseline's accuracy of 77.2 ±0.10% as the Wilcoxon Signed Rank test [44] p-value between the two was measured at 0.07446. This configuration, however, does not reflect the best f1 score.

The best f1 score, 58 ±0.10 %, was achieved by the un-normalised variation of branching entropy mode (VBE) at a training set size of one and half million (1.5 million) words using an unsmoothed 9-gram language model and the *sum* operator. This score is statistically better than the rest of the scores with a maximum pair-wise p-value of 0.0051.

For Normalised Variation of Branching Entropy modes, the sum of the left and right branching measures performed better than the minimum of the two, implying that a smoothing effect is better, as the sum is a form of averaging the two branching directions. Unsmoothed language models performed better than modified Kneser-Ney smoothed language models. This suggests that character level language modelling does not suffer from sparsity, which is prevalent in word level language modelling.

Fig. 3 shows the trend of the segmenters including Morfessor-Baseline and the best performing XBES modes for accuracy and f1 score in relation to training set size. Because the random segmenter was not trained, it is represented as a flat line across the training set sizes.

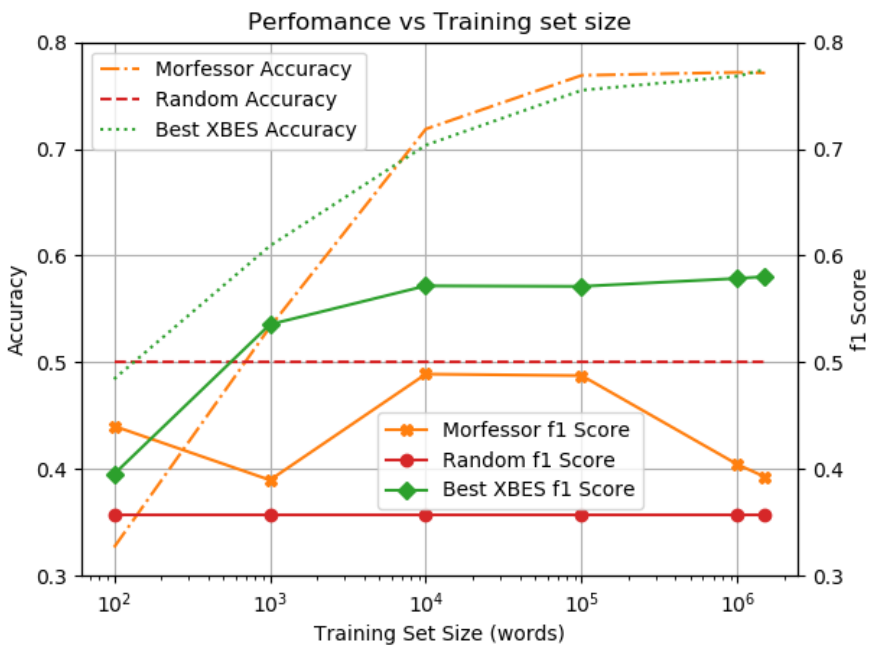

Fig. 3. Average accuracy and f1 score of the Random Segmenter, Morfessor-Baseline and the best XBES mode by training set size

As can be seen from Fig. 3 Morfessor-Baseline's accuracy and the best XBES mode peak at around 77% with maximum training set size. The f1 score is however a different matter.

The f1 score of Morfessor-Baseline peaks at 10000 words and degrades whilst XBES's best mode continues to grow albeit marginally.

6 Conclusions

In this paper, an unsupervised morphological segmenter for isiXhosa that uses branching entropy is evaluated. The IsiXhosa Branching Entropy Segmenter (XBES) uses an adaptation of branching entropy techniques detailed in [40] applied to isiXhosa.

The study contributes and summarises a single bi-directional branching entropy language model with an option for smoothing with modified Kneser-Ney smoothing.

The morpheme boundary identification average accuracy of XBES, at 77.4 ±0.32%, was evaluated to be comparable to Morfessor and it was achieved using the z-score normalised variance of branching entropy mode with an unsmoothed 11-gram language model and the *sum* operator when trained on 1.5 million words.

The morpheme boundary identification f1 score of XBES, at 58 ±0.10%, performed better than the benchmark Morfessor-Baseline when using the un-normalised variance of branching entropy (VBE) mode with an unsmoothed 9-gram language model and the *sum* operator when trained on 1.5 million words.

The results also show that XBES performance could still grow in both accuracy and f1 score, however those gains could cost a lot of training data.

The results also show that using the modified Kneser-Ney smoothing provides no advantage when using branching entropy.

Acknowledgment. The authors thank the South African Centre for Digital Language Resources (SADiLaR) (https://www.sadilar.org) for providing a central source of data and resources for South African Natural Language Processing work.

The authors also with to thank the two anonymous reviewers for their careful reading of the paper and their many insightful comments and suggestions. This has helped to improve and clarify this paper.

References

1. Lulamile Mzamo, Albert Helberg, Sonja Bosch: Towards an unsupervised morphological segmenter for isiXhosa. In: Proceeding of 2019 SAUPEC/RobMech/PRASA Conference. pp. 166–170. , Bloemfontein, South Africa (2019).
2. Sharma Grover, A., van Huyssteen, G.B., Pretorius, M.W.: South African human language technologies audit. Lang. Resour. Eval. 45, 271–288 (2011). https://doi.org/10.1007/s10579-011-9151-2.
3. Statistics South Africa: Community survey 2016 in Brief. (2016). https://doi.org/10.1007/978-3-540-95991-5-71.
4. Moors, C., Calteaux, K., Wilken, I., Gumede, T.: Human language technology audit 2018: Analysing the development trends in resource availability in all South African languages. In: SAICSIT 2018. pp. 296–304. ACM, Port Elizabeth, South Africa (2018). https://doi.org/10.1145/3278681.3278716.
5. Bosch, S., Pretorius, L., Fleisch, A.: Experimental Bootstrapping of Morphological Analysers for Nguni Languages. Nord. J. African Stud. 17, 66–88 (2008).
6. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT

Press (1999). https://doi.org/10.1162/coli.2000.26.2.277.
7. Kosch, I.M.: Topics in Morphology in the African Language Context. Unisa Press, Pretoria (2006).
8. Hammarström, H., Borin, L.: Unsupervised learning of morphology. Comput. Linguist. 37, 309–350 (2011).
9. Zwicky, F.: Entdecken, erfinden, forschen: im morphologischen Weltbild. Muenchen: Droemer. (1966).
10. Louw, J.A., Finlayson, R., Satyo, S.C.: Xhosa Guide 3 for XHA100-F. University of South Africa, Pretoria (1984).
11. Pahl, H.W.: IsiXhosa. Educum Publishers, King Williams Town (1982).
12. Theron, P., Cloete, I.: Automatic acquisition of two-level morphological rules. In: Proceedings of the fifth conference on Applied Natural Langauge Processing. pp. 103–110. Morgan Kaufmann Publishers, San Francisco, CA, Washington, DC (1997).
13. Pretorius, L., Bosch, S.E.: Finite-State Computational Morphology: An Analyzer Prototype For Zulu. Mach. Transl. 18, 195–216 (2005). https://doi.org/10.1007/s10590-004-2477-4.
14. Eiselen, R., Puttkammer, M.J.: Developing Text Resources for Ten South African Languages. In: Calzolari, N., Choukri, K., Declerck, T., Loftsso, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). pp. 3698–3703. European Language Resources Association (ELRA), Reykjavik, Iceland (2014).
15. Mzamo, L., Helberg, A., Bosch, S.: Introducing XGL-a lexicalised probabilistic graphical lemmatiser for isiXhosa. In: Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech) 2015. pp. 142–147. IEEE, Port Elizabeth, South Africa (2015).
16. Nogwina, M.: Development of a Stemmer for the IsiXhosa Language, http://libdspace.ufh.ac.za/bitstream/handle/20.500.11837/221/M Sc %28Computer Science%29 - NOGWINA%2C M.pdf?sequence=1&isAllowed=y, (2016).
17. Déjean, H.: Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In: D.M.W. Powers (ed.) NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Natural Language Learning. pp. 295–299. ACL, Adelaide (1998).
18. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor Variety Criteria for Chinese Word Extraction. Comput. Linguist. 30, 75–93 (2004).
19. Sun, M., Shen, D., Tsou, B.K.: Chinese Word Segmentation without Using Lexicon and Hand-crafted Training Data. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2. pp. 1265–1271. Association for Computational Linguistics (1998).
20. Ye, Y., Wu, Q., Li, Y., Chow, K.P., Hui, L.C.K., Yiu, S.M.: Unknown Chinese word extraction based on variety of overlapping strings. Inf. Process. Manag. 49, 497–512 (2013). https://doi.org/10.1016/j.ipm.2012.09.004.
21. Ando, R.K., Lee, L.: Mostly-Unsupervised Statistical Segmentation of Japanese: Applications to Kanji. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (2000).
22. Rissanen, J.: Modelling by the shortest data description. Automatica. 14, 465–471 (1978).
23. Kit, C.: A Goodness Measure for Phrase Learning via Compression with the MDL Principle. In: Ivana Kruijff-Korbayova (ed.) Proceedings of the Third ESSLLI Student Session. pp.

175–187 (1998).
24. Golénia, B., Spiegler, S., Flach, P.A.: Unsupervised morpheme discovery with UNGRADE. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 633–640. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15754-7_76.
25. Creutz, M., Lagus, K.: Unsupervised Discovery of Morphemes. In: Procedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology. pp. 21–30. , Philadelphia, USA (2002).
26. Gaussier, E.: Unsupervised learning of derivational morphology from inflectional lexicons. In: Proceedings of ACL'99 Workshop: Unsupervised Learning in Natural Language Processing. pp. 24–30 (1999).
27. Chavula, C., Suleman, H.: Morphological Cluster Induction of Bantu Words Using a Weighted Similarity Measure. In: Proceedings of SAICSIT '17,. p. 9. , Thaba Nchu, South Africa (2017). https://doi.org/10.1145/3129416.3129453.
28. Booij, G.: The grammar of words: an introduction to linguistic morphology. Oxford University Press (2012).
29. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. J. Am. Soc. Inf. Sci. Sep. 41, 391–407 (1990).
30. Méndez-Cruz, C.-F., Medina-Urrea, A., Sierra, G.: Unsupervised morphological segmentation based on affixality measurements. (2016). https://doi.org/10.1016/j.patrec.2016.09.001.
31. Belkin, M., Goldsmith, J.: Using eigenvectors of the bigram graph to infer morpheme identity. In: Proceedings of the ACL-02 workshop on Morphological and phonological learning. pp. 41–47. Association for Computational Linguistics, Philadelphia, USA (2002).
32. Schone, P., Jurafsky, D.: Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In: Proceedings of CoNLL-2000 and LLL-2000. pp. 67–72. , Lisbon, Portugal (2000).
33. Goldwater, S., Griffiths, T.L., Johnson, M.: Interpolating Between Types and Tokens by Estimating Power-Law Generators. In: Advances in neural information processing systems. pp. 459–466 (2005).
34. Uchiumi, K., Tsukahara, H., Mochihashi, D.: Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. pp. 1774–1782. Association for Computational Linguistics, Beijing, China (2015).
35. Sirts, K., Goldwater, S.: Minimally-Supervised Morphological Segmentation using Adaptor Grammars. Trans. Assoc. Comput. Linguist. 1, 255–266 (2013).
36. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Pattern Anal. Mach. Intell. 6, (1984).
37. Smith, N.A., Eisner, J.: Contrastive Estimation: Training Log-Linear Models on Unlabeled Data. In: Proceedings of the 43rd Annual Meeting of the ACL. pp. 354–362. , Ann Arbor, Michigan, USA (2005).
38. Narasimhan, K., Barzilay, R., Jaakkola, T.: An Unsupervised Method for Uncovering Morphological Chains. Trans. Assoc. Comput. Linguist. 3, 157–167 (2015).
39. Chen, S.F., Goodman, J.: An Empirical Study of Smoothing Techniques for Language

Modeling. , Cambridge, Massachusetts (1998).
40. Magistry, P., Sagot, B.: Unsupervized Word Segmentation: the case for Mandarin Chinese. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. pp. 383–387. Association for Computational Linguistics, Jeju, Republic of Korea (2012).
41. Zhikov, V., Takamura, H., Okumura, M.: An Efficient Algorithm for Unsupervised Word Segmentation with Branching Entropy and MDL. Inf. Media Technol. (reprinted from Trans. Japanese Soc. Artif. Intell. 2013, 28(3), pp. 347-360). 8, 514–527 (2013).
42. South African Parliament: UMgaqo-siseko weRiphablikhi yoMzantsi-Afrika ka-1996. (1996).
43. Snyman, D., Van Huyssteen, G.B., Daelemans, W.: Cross-Lingual Genre Classification for Closely Related Languages. In: Twenty-Third Annual Symposium of the Pattern Recognition Association of South Africa. pp. 132–137 (2012).
44. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1–30 (2006). https://doi.org/10.1016/j.jecp.2010.03.005.