AfriWeb: A Web Search Engine for a Marginalized Language

Nkosana Malumba, Katlego Moukangwe, and Hussein Suleman

Department of Computer Science
University of Cape Town
Private Bag X3, Rondebosch
South Africa
n.malumba@gmail.com, katlego@moukangwe.com, hussein@cs.uct.ac.za

Abstract. isiZulu is a Bantu language spoken by approximately 9 million people, but with very few written documents available on the Internet. The lack of electronic documents and supporting infrastructure to store and retrieve documents in isiZulu is an additional threat for its survival as a written language. This paper documents an investigation into the creation of one such infrastructural element - a custom Web search engine - for isiZulu, where previously no such system was in existence. The focus of the search engine was on the language-specific elements of morphological parsing and statistical language modelling. Morphological parsing was shown to produce better results for isiZulu, an agglutinative language, than traditional affix-based stemming. Statistical language modelling was able to successfully separate isiZulu documents from others, thus enabling the use of a language-based focused crawler.

Keywords: isiZulu, Web search, morphological analysis, language modelling, focused crawling

1 Introduction

IsiZulu is one of South Africa's 11 official languages. It is the most widely spoken home language in South Africa and is understood by more than 50% of South Africa's population of about 53 million people [1]. In spite of the prevalence of this language in South Africa, it is almost impossible to find information on the World Wide Web written in isiZulu i.e., submitting a query in IsiZulu to a popular search engine, such as Google, and getting results only or mostly in isiZulu.

One reason for this is that digital documents in Zulu are very rare; in October 2014, there were only 682 Zulu-Wikipedia documents [2]. However, the number of such documents has the potential to increase as more speakers of the language become digitally literate, more government documents are produced, more documents on the Web are translated and more books are produced for teaching, learning and popular consumption. Increasing the use of a written language requires a multi-pronged solution to motivate the creation of content. Systems for

storage and retrieval of documents are one aspect of this solution; if a document in isiZulu cannot be found on the Internet, there is little motivation to put it online.

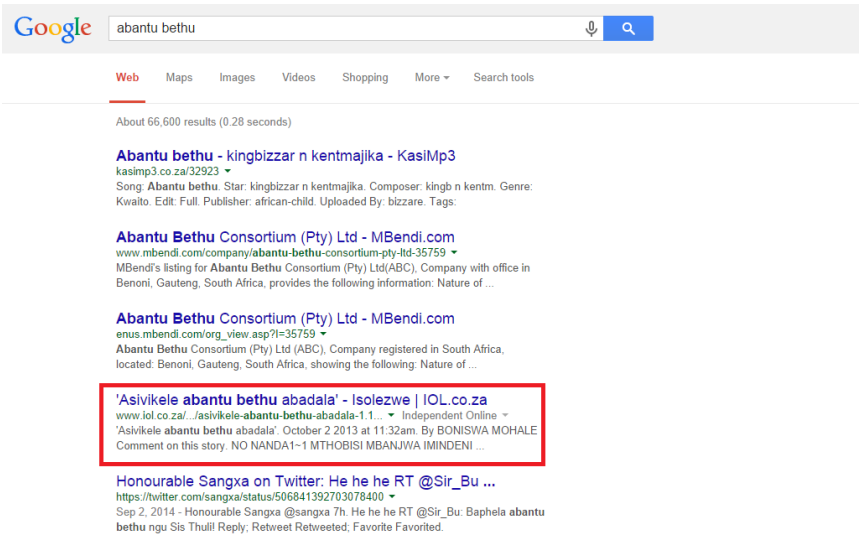

Fig. 1. Google search result for "abantu bethu"

As seen in Figure 1, a Google search for the word "abantu bethu" – meaning: our people – results in the first page with content written in IsiZulu being ranked fourth. The system appears to be biased towards English documents that contain those words surrounded by English content. The small amount of isiZulu content online means that it has little prominence in the networked community and will not be highlighted in link-based search results. Ali [3] showed experimentally how documents in languages other than English are automatically ranked lower because of differences in collection statistics when search engines ignore the language of documents. This anomalous behaviour applies to isiZulu as well.

Therefore, to increase the prominence and discoverability of isiZulu documents online, it is necessary to develop tools and algorithms that are specifically intended to, firstly, create a level playing field for all languages and, secondly, boost the visibility of a language for particular developmental reasons. This paper describes the AfriWeb project that investigated both aspects by testing the viability of search engine algorithms - specifically morphological analysis - for isiZulu and an isiZulu portal with documents crawled from the Web using a focused crawler.

2 Related Literature

Africa has 54 countries that use an estimated total of over 2000 languages. Some languages are endangered due to the assimilation of other dominant groups and the adoption of Western cultures [4]. Language is an element of culture as it presents the philosophy, history, stories and medicinal practices of that particular culture. Therefore, the extinction of a language will inevitably result in the loss of diversity within a larger community [4].

Although there are a large number of spoken languages in Africa, many of these, especially the Bantu languages of South Africa, are among the least researched languages in the world [5]. As a result, technologies that are crucial in the advancement of information retrieval research, such as corpora and dictionaries, are still either undeveloped or incomplete. In the case of IsiZulu, which is the focal language of the AfriWeb project, many researchers in linguistics have provided different perspectives that have resulted in a distributed and non-cohesive body of knowledge [6].

Cosijn et al. [7] conducted the only known study of isiZulu as a language for information access. They considered cross-language information retrieval (CLIR) and analyzed the requirements for and difficulties in developing text processing systems for digital accessibility of indigenous knowledge in isiZulu. Their critique concluded by showing that there are multiple problems and difficulties encountered when implementing CLIR for African languages, including : ambiguity, incorrect stemming, paraphrasing in translations, untranslatability and mismatching [7].

The morphology of African languages has been studied in the context of other languages. El-Khair [8] and Nwesri, et al. [9] highlighted the importance of understanding the characteristics of Arabic to create effective information retrieval systems, especially the need for morphological analysis to handle the highly inflected word forms. Similar morphological analysis approaches were explored successfully by Hurskainen [10] for kiSwahili and Tune et al. [11] for the Ethiopian Afaan Oromo language.

In order to obtain documents for this project, a focused crawler was used; its goal is to selectively seek out pages that are relevant to a pre-defined set of topics (in our case a language) and follow relevant links to crawl through the Web [12] [13]. A focused crawler is driven by a language identification algorithm, based on comparisons of language profiles. Generating a language profile involves breaking the text from the category sample into n-grams and counting the occurrence of each n-gram [14]. Kneser-Key smoothing [15] was used in this study. It provided an additional tuning of simple n-gram counts by smoothing the effect of unigrams with unduly high frequencies because of co-occurrences.

3 General Architecture

The general architecture of the system is indicated in Figure 2.

There are two main parts to the system: the indexing and retrieval of data, and the harvesting of Web pages and documents from the Web. An outline of

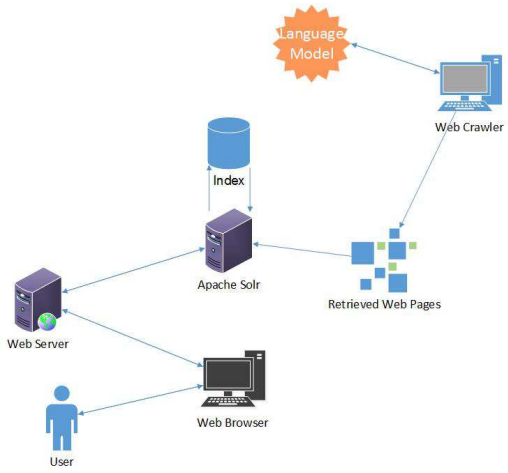

Fig. 2. General architecture of AfriWeb search system

the technologies that were employed in the development of the system are as follows:

- Web Server – required to host the AfriWeb Search Engine application.
- Search Engine Interface – the set of Web pages that the user is going to interact with, hosted on the Web server. Through this interface, a user can submit a query that is based on a particular information need and be able to view the results from the search engine.
- Apache Solr – an information retrieval toolkit, which will primarily be used to index and retrieve documents. SOLR was customized through plugins and alteration of the schema properties to ensure efficient indexing and retrieval of isiZulu documents.
- Focused Web crawler – an application that systematically browses the Web to index the contents of websites that are relevant to some constraint, in this case language of the content. Given a set of URLs as inputs, the crawler visits these URLs and, based on its set of rules, indexes the page and scans for other URLs within the same page, which it can visit next. The crawler was used for harvesting isiZulu text from the Web.
- Language model – a probabilistic model of a specific language, in this case isiZulu. Probabilities indicate an estimate of the likelihood of a given text being in a particular language. In this case, a language model was used to classify a Web page or parts thereof as being written in isiZulu or not.

Many of these components are standard Web search engine components. The language-specific components are the morphological parser that is needed by the SOLR indexer and the model-based language identification that drives the focused crawler. Each of those is discussed and evaluated in the sections that follow.

4 Morphological Parsing

4.1 Design

The relationship between a query and a document is determined to a large degree by the frequency of the query terms in individual documents. However, as documents have to adhere to language constraints, a single word may have morphological variants and the matching algorithms used by search engines will not match the possible variants. Therefore, by reducing the terms to their root form, the results returned by the search engine will have higher probability of relevance. Two algorithms were tested in this process: a prefix/suffix-based stemmer and a morphological parser.

There are two main principles that are used in the development of the stemming algorithm: iteration and longest-match. The iteration principle assumes that affixes are attached to stems in a certain order using a predefined class of affixes. The algorithm simply removes the affixes from either start to end or end to start, based on which class the detected affix matches. The second principle - the longest match - states that within any given class of endings, if more than a single ending provides a match, the longest one should be removed from the word. The affixes used in the development of the stemming algorithm included prefixes from the noun classification system and nominal suffixes. An example of a removed suffix is the diminutive 'ana' that is removed from 'abantwana'.

In the field of linguistics, the morphology of a language is the study of the word formation process in a language based on the parts of the language structure such as morphemes, affixes and other language phenomena that occur in the word formation process. Morphological analysis allows the breakdown of a word into various components that would have been overlooked by light stemming algorithms [16], such as the one described above. Once the semantic structure is obtained from morphological analysis, the parser is then able to apply predefined computations to the word to extract the root word.

The development of a morphological parser focused on the word formation rules that were described by Pretorius and Bosch [5]. The aim of using a morphological parser was to reverse word formation rules (such as ikhanda+ana transforming into ikhanjana) by detecting word patterns based on the affixes and word formation rules. In terms of the prefixal analysis, the scope focused on the noun classification and concordial systems. The noun classification system forms the basis of all prefixes and determines the types of concords that can be applied to a stem. These concords have different categories, which have different semantics that must be considered when analyzing a particular word. Figure 3 is the pseudo-code for the final analyzer that was used.

4.2 Evaluation

Two experiments were conducted.

The first experiment evaluated the accuracy of the morphological parser compared to the stemming algorithm given a language corpus. The experiment was

```
Check the suffix category:
        If (word has a preprefix):
                IF (consonant preprefix AND non-suffix category):
                        CHECK concord class:
                                IF possessive concord:
                                        MARK as root
                                ELSE:
                                        CHECK concord pattern
                                        MARK and REMOVE concord
                                        MARK word as root

                ELSE IF (consonant preprefix AND simple-suffix
category):
                        CHECK concord class:
                                MARK and REMOVE concord
                                MARK word as root

                ELSE IF (consonant preprefix AND special suffix
category):
                        CHECK formation rule for suffix:
                                REVERSE word formation
                                REMOVE prefix
                                MARK word as root

                ELSE IF (vowel prefix and special suffix category):
                        CHECK preprefix class:
                                REVERSE word formation
                                MATCH prefix pattern
                                REMOVE prefix
                                MARK word as root

                ELSE:
                        #word has no preprefix
                        MARK word as root


RETURN root;
```

Fig. 3. Morphological parser algorithm for isiZulu

conducted using the Ukwabelana open source isiZulu corpus [17]. The corpus has a set of 10040 words that have been deconstructed into morphemes and roots. The morphological parser produced 48% accuracy in which the result contained the stem or root word, as compared to 42% produced by the stemming algorithm.

The second experiment required the pre-processing algorithms to be used in the search engine as a pre-processing step during indexing and retrieval of information. 12 Subjects who knew isiZulu were recruited. The purpose of the experiment was to measure if the relevance of the results of a user's query would increase using a morphological parser in comparison to a stemming algorithm. The use of a morphological parser in the indexing and querying of data resulted a in higher precision score as opposed to the stemming algorithm. The mean precision of the morphological parser was found to be 0.138, compared to the stemming algorithm's score of 0.102.

In both cases, the morphological parser resulted in a higher accuracy in the reduction of words, due to its sensitivity to the word formation rules and morphemes used in the derivation of inflected forms. This allows the morphological parser to better deconstruct words using a set of predefined morphemes. In the case of stemming, a brute force stripping of suffixes and prefixes may have resulted in a phenomenon called understemming or overstemming. These processes usually result in the word being incorrectly stemmed due to some of the morphemes either being incorrectly detected, or totally omitted by the algorithm.

5 Language Identification

5.1 Design

Supervised classification is choosing the right label for a given input. The language model was used to decide if a given string was isiZulu or non-isiZulu. Strings were used because documents online are seldom in isiZulu only. The language model was trained using the Ukwabelana sentence corpus [17]. Words in the training data were broken up into n-grams; n-grams such as 'ukw' and 'nhl' are indicative of isiZulu. The language model was also trained using the VariKN language modelling toolkit [18], which included support for Kneser-Key smoothing. Given the language model, a Bayesian classifier, written in Python, was then used to decide if a string belongs to isiZulu or not, by comparing the probabilities of the given text with those of isiZulu and English (as normative language). This language attribute was then used by the focused crawler in deciding whether a page as a whole was mostly isiZulu or not.

Figure 4 is the pseudo-code for the language analysis algorithm that was used.

5.2 Evaluation

The language identification method was tested on 4 datasets.

IsiZulu contained 29423 variable length sentences with 100% Zulu words. *PEnglish* contained 28000 variable length sentences with 100% English words.

```
Algorithm  IDENTIFYLANGUAGEWITHSMOOTHING(input)

global english-pdf,Zulu-pdf,english-dict,Zulu-dict
procedure LOADLANGAUGEMODEL(smoothedARPAfile)
 comment: pdf is a Python defaultdictionary
 pdf ← {}
 for each ngram ∈ smoothedARPAfile
   do pdf[ngram] ← ngram probability
 return (pdf)

procedure CALCULATEPROBABILITY(pdf,word)
 p ← 1
 for each ngram ∈ word
   do p ← p *pdf[ngram]
 return (p)

main
 if input ∈ Zulu-dict and input ∈ english-dict
   then return (non − Zulu)
 if input ∈ Zulu-dict
   then return (Zulu)
 if input ∈ english-dict
   then return (non − Zulu)
 english-pdf ← LOADLANGAUGEMODEL(english-arpa-file)
 Zulu-pdf ← LOADLANGAUGEMODEL(Zulu-arpa-file)
 english-probability ← CALCULATEPROBABILITY(english-pdf)
 Zulu-probability ← CALCULATEPROBABILITY(Zulu-pdf)
 if |Zulu-probability−english-probability| ≤ 10^3
   then return (non − Zulu)
 if Zulu-probability < english-probability
   then return (non-Zulu)
   else return (Zulu)
```

Fig. 4. Language identification algorithm for isiZulu

Table 1. Accuracy for language identification

Dataset	isiZulu(Y)	Non-isiZulu(N)	Non-isiZulu(Y)	isiZulu(N)
IsiZulu	98.4	1.6	0	0
PEnglish	0	0	98.8	1.2
PItalian	0	0	87.6	12.4
PZplus	97.5	2.5	98.3	1.6

PZplus contained 25000 variable length sentences of which 40% was Zulu and 60% was English. *PItalian* contained 25215 variable length sentence with 100% Italian words.

The results from accuracy tests are shown in Table 1. The percentages indicated are for the number of words identified correctly as isiZulu or not (Y) and those identified incorrectly (N). Of the isiZulu documents, 98.4% were correctly identified. Most documents in English or Italian were correctly identified as non-isiZulu; Italian had understandably lower accuracy, as the system was trained with English n-grams. The mixed documents had 97.5% accuracy for the isiZulu subset.

Some Zulu words were classified as non-Zulu because they also appear in English (into, bake, etc.). A small number of words in the Zulu corpus were also found to be English words for which there was no equivalent in Zulu (e.g., assurance).

6 Conclusions

This project centred on a Web search engine to support the development of content in a marginalized language – isiZulu. In attempting to meet this objective, language-specific algorithms were developed for morphological analysis of the language and for identification of the language within a focused crawler.

The morphological parser achieved an accuracy level of 48%, which surpassed the accuracy level of a typical affix-driven stemmer. However, there is clearly scope for further investigation on stemming in Bantu languages. The statistically-driven language identification was mostly successful, with isiZulu documents being separated from non-isiZulu documents with an accuracy greater than 90%. When used with a focused crawler, more than 60000 documents were successfully obtained.

The short supply of text resources, formal language grammars and foundational work in isiZulu computational linguistics made this project particularly difficult. However, it was shown that specific components of a Web search engine can be optimized for a marginalized language, with some success. It is hoped that the availability of such search engines can lead to greater interest in written documents in isiZulu, feeding back into the development of better language tools, in a cycle that ultimately promotes and preserves marginalized African languages. The number of speakers of isiZulu is not decreasing over time and there is greater recognition of national languages in South Africa so this work

will support an expanding community of writers and readers of an important language.

Acknowledgement. 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. Statistics South Africa: Census 2011, 2012. http://www.statssa.gov.za/census2011/default.asp
2. Wikipedia: Ikhasi Elikhulu, Wikimedia Foundation, 2014. http://zu.wikipedia.org/wiki/Ikhasi_Elikhulu
3. Ali, M. M., Suleman, H.,: Mixed Language Arabic-English Information Retrieval. In 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2015), pp. 427–447, Cairo, Egypt, 14-20 April 2015, Springer (2015).
4. Mukami, L.: Africa's endangered languages. African Review (2013) http://www.africareview.com/Special-Reports/Africas-endangered-languages/-/979182/2008252/-/12yos0s/-/index.html
5. Pretorius, L., Bosch, S.E.: Finite-state computational morphology: An analyzer prototype for Zulu. Machine Translation. 18(3), pp. 195–216 (2003)
6. Madondo, L.M., Muziwenhlanhla, S: Some aspects of evaluative morphology in Zulu (2000)
7. Cosjin, E., Pirkola, A., Bothma, T., Jarvelin, K.: Information access in indigenous languages: a casestudy in Zulu. In South African Journal of Libraries and Information Science, 68(2), p. 94 (2002)
8. El-Khair, l. . A.:Arabic Information Retreival. In Annual Review of Information Science andTechnology. Egypt: John Wiley and Sons, pp. 505–533 (2007)
9. Nwesri, A.F., Tahaghoghi, S.M., Scholer, F.: Answering English Queries in Automatically Transcribed Arabic Speech. In 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), IEEE (2007)
10. Hurskainen, A.: Swahili Language Manager. In Nordic Journal of African Studies, 8(2), pp. 139-157 (1999)
11. Tune, K. T., Varma, V., Pingali, P.: Evalutation of Oromo-English Cross Language Information Retrieval, Hyderabad, India, Cross Language Evaluation Forum (2007)
12. Chakrabarti, S., van der Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific Web resource discovery. Computer Networks. 31, pp. 1623–1640 (1999)
13. Novak, B.: A survey of focused web crawling algorithms (2004)
14. Cavnar, W. B., Trenkle, J. M.: N-gram-based text categorization. In 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR 94), pp.161–175 (1994)
15. Chen, S. F., Goodman, J.: An empirical study of smoothing techniques for language modeling. Computer Speech & Language. 13(4), pp. 359–393 (1999)

16. McEnery, T.: Corpus linguistics: An introduction. Edinburgh University Press (2001)
17. Spiegler, S., Van Der Spuy, A., Flach, P.A.: Ukwabelana: an open-source morphological Zulu corpus. In 23rd International Conference on Computational Linguistics, Association for Computational Linguistics (2010)
18. Siivola, V.: VariKn - Language modelling toolkit (2007) http://forge.pascal-network.org/docman/view.php/33/58/variKN_toolkit.html