Towards Adaptation of NLP Tools for Closely-Related Bantu Languages: Building a Part-of-Speech Tagger for Zulu

Mariya Koleva

Supervisors

Professor Dietrich Klakow

Dr Alexis Palmer

Abstract

Standard machine learning approaches in NLP require large amounts of data for training. These do not exist for the majority of languages. Creating annotated data, which is crucial for supervised approaches, can be both expensive and time-consuming. The result is that languages for which such data is missing may not receive the attention of the language technology community. This thesis addresses the question whether it is possible to build an accurate NLP tool for a low-resource language using small amounts of data and some linguistic information. It also investigates whether such a tool can be further adapted to perform on a genetically related language by harnessing the power of crosslingual similarities between both languages.

We use an off-the-shelf NLP toolkit (OpenNLP) to train a number of models for the morphological analysis and part-of-speech (POS) tagging of Zulu - a South African Bantu language of the Nguni group with 10.3 million speakers. The training is done with a linguistically informed semi-supervised approach, where each POS model is incrementally augmented with different features derived by our linguistic knowledge of Zulu. The models are then tested and compared. Some implications are then discussed, having to do with the possibility to linguistically adapt and apply the best-performing model to another, closely related Bantu language.

Contents

1 Introduction

Africa is a continent which is linguistically very rich, and it is the home of many ethnicities and many typologically diverse languages (the count of the latter is estimated to be around 2,000). In the Republic of South Africa alone, there are 11 official languages, of which nine are indigenous Bantu languages with a large number of speakers. Bantu languages are among the better-researched languages in African linguistics. In terms of language technology development, though, they are lagging behind and fall into the group of low-resource languages.

There is no single reason why a language would be under-resourced. An endangered language or one with a small number of speakers is often resource-poor, but there is no one-to-one mapping between the number of speakers of a language and the availability of resources for it. The South African Bantu languages are widely spoken (with for instance 10.3 million speakers of Zulu and 7.9 million of Xhosa) but due to the social and political past of South Africa they did not become strategic in education and commerce until recent years. The resulting lack of linguistic resources, both primary (e.g. text and spoken corpora, wordlists) and secondary (e.g. dictionaries, grammars, study guides), and tools (e.g spell-checkers) holds back the development of African language technology. Even though Zulu (the language of choice for this study) is an official language, and its use is encouraged and aided in different ways, the fact remains that English is the dominant language of trade, public life and education (Spiegler et al., 2010a).

Researching low-resource languages is worthwhile for more than one reason. Academically, in-depth research of lesser-known languages may give insight on linguistic phenomena that are not encountered in better-researched languages such as English, deepening our understanding of the capabilities of human language. African languages are a prime example for this, as many of their typological characteristics are distinctly different from those of other language families of the world.

From the point of view of society, resulting language technology (LT) applications may either benefit people directly, e.g. in the form of spell-checkers and translation systems, or indirectly through supporting and encouraging further research. NLP tools can for example motivate native speakers with either specialised knowledge or native competence to participate in the research effort (Childs, 2003, p12).

Typological differences are not the only interesting research point of African languages. Due to the strong social and cultural diversity, the historical contact with speakers of other languages during colonialism, and also because of increasing urbanisation and migration, the linguistic situation in African states is very dynamic. On the one hand, African languages play a big role in the formation vernacular languages of the New World (Childs, 2003, p.8)). On the other hand, new urban varieties of indigenous languages develop (e.g., "Town" vs. "Red", or traditional, Xhosa (Childs, 2003, p.8)). In South Africa, code-switching and linguistic borrowing between English, Afrikaans and the indigenous languages is widespread, giving rise to new research questions both for general linguistics and for the natural language processing (NLP) community.

A decision that the language technologist needs to make when dealing with any language, including low-resource languages, is how language-specific a solution to pursue. Language-independent solutions, typically probabilistic, require a lot of data. When data is available, probabilistic approaches are quick and inexpensive. Language-specific solutions rely more on manually defined rules and constraints. Such rule-based approaches can perform well for tasks with a limited scope. However, they don't scale well, as developing rules is expensive and human intuition often translates badly into machine logic. Rule-based systems also tend to be prescriptive in nature, rather than descriptive, i.e. they describe how a language *should* be used, as opposed to how it *is* used by speakers. Consequently, these systems perform poorly when faced with noisy real-world data (De Pauw and de Schryver, 2009)

Scalability also depends on the task in question. While formalizing a language's entire syntax and handcrafting machine-readable rules for it can be very arduous and time consuming, modelling the morphology of a language is often a lot more feasible (Roark and Sproat, 2007). Still, the development of a mature and robust morphological analyzer for a morphologically complex language is a lengthy and involved process. ZulMorph, the UNISA protoype morphological analyzer for Zulu, took a decade to develop (Bosch et al., 2008a).

For such cases were scalability is an issue, a data-driven approach may be beneficial, despite the previously mentioned issues. NLP offers many techniques for processing and generalizing from large amounts of data. When data is limited, there are still a number of machine learning approaches that can be employed in order to achieve satisfactory solutions. With some linguistic expertise in the form of, for instance, understanding the phonological, morphological, and syntactic structure of a given low-resource language, standard machine learning approaches could be tailored to the needs of low-resource NLP.

2 Zulu and the Bantu Languages

This section presents a simplified overview of the Bantu languages, specifically their morphology. It focuses on features common to the Bantu group as such - the Bantu languages are tied closely together in terms of phonology, morphology, syntax, and vocabulary - as well as on some features of the Niger-Congo family. We are primarily interested in generally applicable features, as the objective of this thesis is to contribute to approaches for low-resource languages which are both linguistically motivated and yet not strictly language-specific.

2.1 Classification

The classification of African languages has been influenced both by geographical factors, as well as by typological ones, and has been the subject of many controversies (Childs, 2003, p.18). Currently, African languages are classified in four phyla, or families: Niger-Congo, Nilo-Saharan, Afroasiatic, and Khoisan. Niger-Congo is the largest phylum and it covers, among many others, the Bantu languages, whose number is nearly 500 (De Pauw et al., 2012). Zulu is a major representative of the latter, together with other widely spoken and researched languages such as Swahili, Youruba, and Hausa. Zulu (also, known as isiZulu) belongs to the Nguni group of South-eastern Bantu languages (Pretorius and Bosch, 2009).

2.2 Writing System

The orthographic system is important to consider when building NLP tools for Bantu languages, especially when considering exploiting their similarities. Zulu and the other three Nguni languages are written conjunctively: the morphemes are concatenated to each other to build word forms. Other languages use a disjunctive system of writing, where some morphemes are written separately from the stem, though still forming one lexical item. Languages from different groups fall onto different points on the spectrum of conjunctive and disjunctive writing systems, which has implications for the development of NLP tools for them. Taljard and Bosch (2006) discuss different approaches to word class tagging (POS tagging) with respect to these differences.

An additional issue for the automatic processing of written Bantu languages is that they often use a larger or smaller set of diacritics to denote some language-specific sounds, such as the clicks for which these languages are popularly known. Linguistic texts may additionally use diacritics to denote tone: Zulu and some other Bantu languages are tonal languages, and in speec, tone would be an important feature for disambiguating what would otherwise (in print) be ambiguous morphemes. The problem of text corpus normalisation and diacritic restoration from a machine learning perspective has been discussed by De Pauw and de Schryver (2009) (see chapter 3).

2.3 Morphology

Zulu and the other languages from this group are agglutinative: a surface word form is composed of concatenated morphemes which encode not only inflectional and derivational information, but some syntax as well. The two most complex categories are the nouns and the verbs, both interacting with each other and the other lexical categories with a system of concords.

2.3.1 Noun Classes and Concord

The Zulu noun is composed of a prefix (which can often be further analysed) and a stem which belongs to one of twelve classes (labeled 1-7, 9, 10, 11, 14, and 15). There are stems which can be observed in more than one class, however; this is especially the case because the singular and plural are separate classes. As Welmers (1974, p. 161) observes about the Bantu:

> “(it is) difficult to say that a particular stem ’belongs in a particular noun class. Rather, a given stem ’occurs’, along with many other stems, in conjunction with a particular prefix or pair of prefixes, and perhaps also with other prefixes or pairs of prefixes as well. The classification is not inherent in noun stems as such, but is rather associated with the prefixes.”

Generally, Zulu noun classes can be identified from the prefix of the noun, and they affect the form of the other parts of speech. The latter phenomenon is known as concord

noun class	example	noun class	example
1	u-mu-ntu ’person’	2	a-ba-ntu ’people’
	u-0-baba ’father’		o-baba ’fathers’
3	u-mu-zi ’village	4	i-mi-zi ’villages’
	u-0-nogwaja ’hare	2	o-nogwaja ’hares’
5	i-0-gama ’name’	6	a-ma-gama ’names’
7	i-si-tsha ’dish’	10	i-zi-tsha ’dishes’
9	i-m-pala ’impala’	10	i-zim-pala ’impalas’
	i-0-khwaya ’choir’	6	a-ma-khwaya ’choirs’
11	u-0-phondo ’horn’	10	i-zim-pondo ’horns’
14	u-bu-hle ’beauty’		
	u-0-tshani ’grass’		
15	u-ku-dla ’food’		

Table 2.1: Noun classes in Zulu, after Spiegler et al. (2010b)

or agreement. Table 2.1 appears in Spiegler et al. (2010b) and presents the Zulu noun classes.

As was mentioned in the previous paragraph, the noun prefix is not always simple. The class prefix, which is concatenated with the stem, can be preceded by a number of other prefixes, which differ depending on the parts of speech with which the word is in agreement. Bantu languages also allow for the concatenation of class prefixes themselves.

> “There are numerous instances in Bantu languages of a single noun form which includes two, or even three or four class prefixes in sequence. In such cases, a stem with one prefix is taken as a ’base’, and to this entire base a further prefix is added; such a form may then again be treated as a base to which yet another prefix is added.” (Welmers, 1974, p.168)

It is also not the case that prefixes and stems are simply concatenated together. Many prefixes have allomorphs which occur in specific contexts, for example before stems which begin with a vowel (Welmers, 1974, p.166). The morphophonemic rules which govern the alternations can be as simple as vowel harmony and “alliterative concord” (Welmers, 1974; Childs, 2003).

2.3.2 Verbs and Concord

The verbs in Bantu language take the following form (taken from Childs (2003, p.104)):

(NEG–) SM – TMA – (OM–) Root (–Ext1-Ext2...) – FV

The verb root, which in Zulu can be as small as a single letter, is preceded by:

- SM - a subject marker,
- TMA - prefixes for tense, mood, and aspect,
- OM - an object marker.

It is followed by:

- Ext1-Ext2 - verb extensions
- FV - final vowel.

This is a very abstract picture, however, and the system of a specific languages can depart from it. Zulu for example also has suffixes, such as the negative suffix or the subjunctive suffix, in addition to the negative and the subjunctive prefixes.

The verbal prefixes SM, TMA, OM, and the optional NEG have grammatical functions and realize the concord between the parts of speech. Thus, in the Zulu verb form *akazi*, *a*- is the negation, -*k*- is the negative subject prefix for noun class 1, and -*azi* is the verb root. -*k*- indicates that the subject is a noun from class 1. In contrast, the related verb form *abazi* agrees with a subject belonging to class 2, as indicated by the prefix -*b*-. The negative prefix and the verb root are the same.

The extensions (verbal suffixes) affect the semantics of the verb root, and are in turn affected by the root semantics, as well as by other extensions that are used together (Welmers, 1974; Childs, 2003). Depending on the Bantu language, the number of concatenated extensions can vary, but Welmers (1974, p.339) reports that up to four extensions are sometimes acceptable - all modifying the meaning and the semantic force of the others to a very nuanced effect. Zulu has six extensions: applied, causative, intensive, neuter, passive, and reciprocal. Zulu verbal forms with one or two extensions are common. Like other morphemes, their form can be affected by morphophonemic rules (Welmers, 1974, p.337).

Finally, as it is the case with noun bases, compositional verb bases are possible in Bantu languages. Such a base is formed by a root and one or more bound morphemes: inflections or other verb roots (Welmers, 1974, p.329).

3 Related Work

3.1 Approaches to low-resource languages

Computational linguistics for low-resource languages is a complex area, because the individual linguistic situations can vary widely. Depending on the endangerment status of a language and its political and commercial importance, projects and priorities vary: they may be as broad and fundamental as data collection, data preservation, and language documentation, or as specific and time-sensitive as machine translation for crisis-handling (Lewis et al., 2011). In recent years, low-resource languages have become the focus of a number of major conferences, such as COLING 2012 and ACL 2013.

Streiter et al. (2006) addresses the differences between projects for high-resource and low-resource languages from a functional perspective. Some of the contrasts he draws are well-known: chronic shortage of data, loss of data because of legacy formats or fragile storage options such as CDs, loss of access when the main researcher changes projects, institutions, or stops working. This section echoes Bird and Simons (2003), who thoroughly review the effects that legacy formats have on language documentation - namely, that painstakingly collected data gets lost because different scholarly communities use incompatible software and formats. Where Bird and Simmons discuss best practices to ensure data portability in general, Streiter outlines ways for small-language developers to ensure that their data and research artefacts survive the end of the project - incorporating language packages into open-source software, and taking advantage of the various licences under the “copyleft” paradigm.

What is particularly informative in Steiter’s presentation is that he discusses at length the process of choosing a research paradigm and what decisions researchers in high-resource and low-resource language projects need to make. “Big-language” projects set trends in research and owing to the plethora or resources can and are expected to experiment with cutting-edge (or simply fashionable) approaches. “Small-language” projects have to maximize their profit from the limited data they have, meaning that they might have to settle for approaches that might have fallen out of favour. Work in high-resource language projects is usually partitioned between different researchers, experts in their discipline; work in low-resource language projects is performed by one or two scholars who cover all positions - language expert, developer, annotator, etc.

3.2 Computational Approaches to Morphology

3.2.1 Finite-state Approaches

Roark and Sproat (2007) offer an overview of the history of computational morphology. A significant portion of their account is focused on the finite-state approaches, which they identify as the dominant paradigm in the discipline (Roark and Sproat, 2007, p.113). They make a case for this aproach by pointing out that there are ways for finite-state tools to handle even difficult cases of non-local dependencies.

In these approaches, an algebraic model of morphology is built. What can be modeled is morphotactics, as well as rules which govern morphophonemic alternations, spelling changes, or morphosyntactic behaviour. A system which has been of fundamental importance to finite state morphology is Koskenniemi (1983)'s *KIMMO Basics*, where "each transducer reads the lexical and surface 'tapes' simultaneously"(Roark and Sproat, 2007, p.104)

3.2.2 Machine Learning Approaches

Machine learning approaches to morphology aim to induce morphology from corpora. Supervised approaches involve presenting the learner with forms and patterns which are known to be related. Unsupervised and semi-supervised approaches attempt to induce morphology from raw corpora. More specifically, they are used to discover morphologically related forms, or to infer through a process of generalization the rules for generating new forms that are not present in the training data (Roark and Sproat, 2007, pp.117-119).

(Roark and Sproat, 2007, p.118) point out that "[...]statistical n-gram language-modelling approaches to morphology have been mostly restricted to agglutinative languages." They review the following two papers on machine learning for agglutinative languages, which relate closely to our methodology.

Hakkani-Tür et al. (2002) present a solution for morphological disambiguation for Turkish - a language with highly agglutinative morphology which leads to large morphological tagsets. Inspired by statistical POS-tagging techniques, they segment the word forms as if they were sentences and the morphemes were words. Then they use SRILM to model trigram probabilities for the new corpus and tagset.

The second paper on n-gram models which Roark and Sproat present is Lee et al. (2002). It models syllable trigrams for Korean, assuming "that syllable trigrams are indicative of part-of-speech tags" (Roark and Sproat, 2007, p.117). Roark and Sproat comment that this treatment of the problem for agglutinative languages is very apt, because morphemes in these languages "obey 'word-syntactic' constraints".

3.3 Computational Morphology and the Low-resource Languages

Creutz and Lagus (2005) report on a morpheme segmentation and morphology induction programme - Morfessor. It is language-independent, which is an advantage when dealing with highly inflective languages such as Finnish and Turkish (which usually require language-specific morpheme analysers). The Morfessor employs a greedy search algorithm to find a "morph" lexicon and to choose the most probable morphological segmentation.

Biemann (2009) proposes an approach to unsupervised POS tagging that differs from earlier approaches in that the tagset is acquired automatically from unstructured text. This is achieved with a graph clustering algorithm and the resulting categories may be different from those suggested by linguists. It is also different in that not every word is necessarily tagged.

Petrov et al. (2012) contribute to the body of research on POS tagging for low-resource languages by proposing a universal POS tagset, which is evaluated on an unsupervised grammar induction task. The tagset consists of 12 coarse-grained categories, which leaves language experts the option to refine the categories and revise the decisions made by Petrov and colleagues when the task calls for it.

3.4 The South African Situation

While there are many African languages which are resource-poor, there is an increase in the computational research going on in the continent. Information about the ever-growing pool of tools, resources, and academic texts is being collected in the `Aflat.org` initiative. South African language technology in particular has been endorsed by the country's government as a priority domain. In 2010 this prompted a country-wide audit of the state of Human Language Technology (HLT).

The outcome of the South African HLT audit is reported by Grover and Huyssteen (2010), who conducted a survey on the technological representation of each of the official South African languages with the purpose of giving recommendations for the future development of South African HLT. It is a qualitative review of for-profit as well as academic institutions. The authors surveyed different organisations about the type of LT components that are available, their maturity, accessibility, and the range of languages they cover. The results show that among the indigenous languages, Zulu and Xhosa are the ones on which work has been more intensively conducted. Some possible reasons for this are the increasing awareness of the large community of speakers and the importance of these languages for trade and education. What also became apparent through this survey, though, is that while there is work on basic components for text pre-processing and morphological analysis for indigenous languages, work on higher-level components is still limited.

Bantu languages are relatively well-researched from the perspective of general linguistics. Because of their complex agglutinating structure, they favour computational research within the rule-based paradigm - intensive work on South African Bantu languages with the Xerox xfst tool (exemplified by the corpus of work by Professors Sonja Bosch and Laurette Pretorius at UNISA), as well some initial work on Swahili and Tswana with the GF tool[1]. While models for some of the languages are limited, the Nguni group is very well represented. There is a mature morphological analyser for

[1] `http://www.grammaticalframework.org/lib/doc/status.html`, December 2012

Zulu (Pretorius and Bosch, 2003) and multiple developments of HLT components for the other languages of the Nguni group (Bosch et al., 2007, 2008b).

In attempts to alleviate the resource situation for South East Bantu languages, Bosch, Pretorius and colleagues research the possibilities for quick development and deployment of morphological analysers for the Nguni languages, to which Zulu belongs. In Bosch et al. (2008a) they describe an experiment in bootstrapping morphological analysers for Xhosa, Swati and Southern Ndebele on the basis of the existing Zulu morphological analyser ZulMorph. ZulMorph consists of two modules: one specifying the existing affixes, roots and permitted combinations, and one for the morphophonological alternations. The authors define bootstrapping as the iterative process of applying ZulMorph to the target languages, analysing the errors, updating the analytical rules and lexicon (where relevant), and analysing the development corpora of the target languages again. At the first iteration ZulMorph is initially applied to a small 200-type parallel corpus. The resulting errors stem mainly from unknown verb and noun roots, as well as incongruent word structures. Next, a Xhosa lexicon is added, which supplies additional types for the use of the Xhosa prototype as well as the the other target languages, and ZulMorph is applied again. At the next step, all the missing roots from the first step are added. Finally, additional rules for the target languages are introduced.

As Streiter pointed out, evaluation for low-resource languages (especially for hybrid systems) is particularly difficult. Firstly because data is insufficient, and secondly because of the novelty of the methods there may not be informative evaluation standards (such as the BLEU metric for machine translation, for instance). Nevertheless, Pretorius and Bosch (2009) succeed in designing a reliable evaluation process. They test the prototypes on a 7000-type parallel corpus and compare the results to the ones from the development corpus. The authors observe a decrease in performance (20%-40%), which they attribute to new roots, named entities, and foreign language borrowings. They conclude that while the rule-based modelling of ZulMorph was time-consuming and challenging, subsequently adapting the morpological analyser for Xhosa, Swati and Southern Ndebele was considerably quicker, and while tests on accuracy are still in progress, the results are promising.

De Pauw and de Schryver (2009) make a case for corpus-based data-driven approaches for African languages. The picture they draw for the language technology situation in

Africa is one of a great mismatch between number of speakers and available language technology resources, Swahili being one of the radical cases with 50 million speakers and a limited number of resources. Another detail of this picture, though, is increased Internet usage in the urban areas, the web being a huge resource of text data available to be gathered through techniques such as web crawling and web scraping.

De Pauw and De Schryver explore data-driven approaches for various African languages, including some South African Bantu languages, and conduct three experiments with limited data, which aim to assess the potential of standard machine learning approaches for low-resource African languages. The criteria for assessment are language-independence, development speed, robustness, and empiricism, tested on various tasks ranging from corpus normalisation to morphological anlysis and part-of-speech tagging. The results show that even though having more data is preferable, accurate solutions can be achived quickly with moderate amounts of data and can be evaluated with real-life (performance) data.

The experiment which is relevant to our current project is the first experiment of De Pauw's and De Schryver's: corpus normalisation. The research question is whether diacritics lost during corpus collection can be restored from the orthographic context. The authors set up a machine learning task, the training data for which is a corpus with correct orthography. From this they identify the pairs of graphemes to be disambiguated. Then they remove all diacritics from the text and form classification features - each character and its immediate context. The features are paired with the correct classes, thus forming the training data for a memory-based learner. During the diacritic restoration task, words from the test data are split and reanalysed into features in the same way and are compared to the features in the training data. The class of each entry in the memory which is the closest to the corresponding test data feature is then assigned to the training feature. This approach was applied to seven African languages from three different families, including the South African Bantu languages Northern Sotho and Venda. In all cases, the memory-based approach beat the baseline, a simple lexicon lookup of words.

De Pauw et al. (2012) also report on two very promising experiments on knowledge-light approaches to the tagging of Bantu languages. They start from the premises that machine learning approaches to POS tagging rely on the context of the token in order to disambiguate and tag it, and that there is little research on the linguistic features - what kinds of features there are, how they are selected, and what exactly is their contribution to the disambiguation task. The authors set up two machine learning experiments for the tagging of four Bantu languages: Swahili, Northern Sotho, Zulu, and Cilubà. The first experiment is on context-based tagging, for which they use a memory-based tagger. The tagger employs two classifier for handling the known and the unknown tokens, respectively. The two taggers use different sets of features. In the case of Zulu, the classifier for known words takes into account the left context of a token. The classifier for unknown words takes into account the suffixes, the prefixes, the token, and its left context. All the features are acquired automatically, because they are strongly dependent on the dataset. While the memory-based tagger performs very well for Swahili, Northern Sotho, and Cilubà, its performance for Zulu is decreased because of the many unknown words.

What is particularly encouraging, however, is the second experiment which uses a Maximum Entropy classifier with a “Bag of Substrings” approach. This experiment has the significant advantage that it does not necessarily require a morphologically annotated corpus for training. Instead, the words are reanalyzed into all their substrings and fed as features to the classifier. The classifier itself figures out what the salient features are. Practically, the authors explain, this is a unigram tagger, but the unigram is supplied as a series of substrings and the unigram class. The results for Zulu are very good. This experiment also serves as a good point of comparison for the current project, since we also approach the construction of a POS tagger from the point of view of morphological analysis and feature selection.

4 Methodology

4.1 The Part-Of-Speech Tagging Task

Part-of-speech tagging is a task for which we need a solution when building NLP tools for low-resource languages. In the classical task of POS tagging, the input consists of a tokenized corpus and a predefined tagset. The process of disambiguation and tagging is done by a tagging mechanism (Jurafsky and Martin, 2008). Two of the most common machine learning frameworks for POS tagging are Hidden Markov Models (HMM) and Maximum Entropy (MaxEnt) models.

4.1.1 HMM tagging

HMM is a probabilistic sequence classifier. It infers the correct tag for a token in a sequence, computing the probability from limited information - it cannot observe all tags in a sequence. The model operates on two assumptions:

1. the probability of a state depends only on the previous state, and
2. the probability of an output observation depends only on the state that produced the previous observation and not on any other state or observations.

The process involves a forward algorithm that computes the likelihood of an observation sequence, a decoding task (commonly performed by the Viterbi algorithm), and learning - performed by the Forward-Backward algorithm.Jurafsky and Martin (2008)

4.1.2 Maximum Entropy

Maximum Entropy (MaxEnt) is a framework whose task is to choose such a classification, that is consistent with a predefined set of constraints, while making the fewest assumptions about the data. In a POS tagging task the classifier extracts a set of features from the input, where each feature has a weight that signals how strong the cue is. The intuition of maximum entropy is to build a distribution by continuously adding features (Jurafsky and Martin, 2008).

4.1.3 Our Approach

Our approach is related to the “Bag of Substrings” approach that De Pauw et al. (2012) take. We adopt the Maximum Entropy framework for POS tagging and explore different configurations for features using substrings in addition to n-grams. The departure point is that we do not use mere orthographic substrings. Instead, we train a morphological analyser which supplies us with linguistically-plausible substrings.

We define “morphological analyser” as a pair of two related model: Morpheme Detector and Morpheme Labeler. A morpheme detector is a model which is applied on tokenized word forms. It detects the morpheme boundaries on the basis of statistical inference from training data, and outputs the word forms segmented into (predicted) morphemes. A Morpheme Labeler is a model which has learned a set of morphological labels (a labeling scheme) and relationships from morphologically analyzed data and labeled data. It is a sequence labeler like a POS tagger, and it labels segmented wordforms (from the output of the Morpheme Detector) with the labels from the labeling scheme.

We propose standard POS tagging features (a token and a context window), simple analyzed tokens (or unigrams in the sense of De Pauw’s and de Schryver’s), analyzed and labeled tokens, and combinations of all. We also suggest strategies for future cross-lingual adaptation.

4.2 Tools

OpenNLP Toolkit

This project does not intend to implement new machine learning algorithms or applications. Instead, the OpenNLP toolkit[1] was used. It is a collection of machine learning tools which allow for building models for various NLP tasks, for applying the models on data, for evaluation and k-fold cross-validation. The model builders are based on a Maximum Entropy implementation, but the POS tagger also allows for training a perceptron model. We have used these tools and their output as a black box, only tuning

[1] `http://opennlp.apache.org/`

for cutoff (the minimum number of times a feature needs to be seen by the learner before it considers it in the training) and number of iterations.

OpenNLP MaxEnt Classifier

In addition to specialized tools, OpenNLP also includes a general Maximum Entropy classifier. This classifier accepts custom-built features and can be used for a large variety of prediction tasks. We used this implementation to train models with linguistically-informed features. Additional tuning for cutoff, iterations, and smoothing is possible, though we only experimented with the former two.

MaxEnt Evaluation Script

As the OpenNLP POS Tagger Evaluator tool is not compatible with the MaxEnt classifier, we needed a separate evaluation script to evaluate the output of the classifier against a gold standard. We based our script on the source code for the OpenNLP POS Tagger Evaluator, evaluating for tag accuracy.

Further Text Processing Tools

For basic text processing we used custom Python and Perl scripts, shell scripts, and the tools in text editors.

4.3 Data

Three corpora were used in this project: the Ukwabelana corpus was used for the training and testing of advanced models for morphological analysis and for POS tagging; a corpus from Wikipedia was used to train basic models for sentence detection and tokenization; the CTexT parallel corpus was processed with the purpose to be used later on in a cross-lingual adaptation task, as well as for the future extension of the adapted system via domain adaptation.

4.3.1 Corpora Description

Ukwabelana Corpus

This is an open-source morphological Zulu corpus consisting of 10,000 manually labeled word types, 100,000 word types, 3,026 POS-tagged sentences, 30,000 raw sentences, a morphological grammar of Zulu in DCG format, a parsing algorithm for identifying possible word roots, and a POS-tagger which "assigns the category to a morphologically analysed word type." (Spiegler et al., 2010a) It is comprised of fiction novel texts and the Zulu translation of the bible. The corpus comes preprocessed without capitalisation or punctuation, one sentence per line. The DCG is a format for grammars in Prolog. The grammar and the parsing algorithm provided were used by the authors during the annotation phase of the corpus creation.

CText Corpus

A proprietary corpus of parallel texts in ten South African official languages (excluding English), collected from the official website of the South African government - South Africa Government Online[2]. The Zulu part of the corpus contans36 838 tokens. The corpus is collected by CTexT[3].

Wikipedia (Wiki) Corpus

A toy corpus of 53 raw sentences (one per line), manually collected from the Zulu Wikipedia[4].

4.3.2 Data Split

The three corpora outlined in the previous section are used for distinct tasks in the experiments. The basic data is split into a training set and a test set.

Sentence Detection

For the Sentence Detection model, the entire Wikipedia corpus was used as a training set. The model was tested on a random subsection of the CTexT corpus. Due to the simple nature of this model, no further divisions of the corpus were necessary.

Morpheme Labeler

A modified version of the morphologically analysed wordlist from the Ukwabelana corpus was used for the Morpheme Labeler. The modification consisted of the following: where the Ukwabelana wordlist gives alternative analyses for a word (listed on the same line,

[2] www.gov.za

[3] http://www.nwu.ac.za/export/sites/default/nwu/p-news/pm_808_a.html

[4] http://zu.wikipedia.org/wiki/Ikhasi_Elikhulu

separated with a comma), we took each analysis as a new corpus entry and transferred it to a new line. This way, the wordlist expanded from 10,000 to 10,188 entries (lines). Two-thirds of the list (6,792 randomly selected lines) formed the training set, which will be referred to as the Original Labels Training Set. One third (3,396 lines) formed the development set, to be referred to as the Original Labels Development Set. The two sets were further modified by replacing the original morpheme labels with the labels from the conflated labeling scheme. The resulting training and development set are thus to be referred to as the Conflated Labels Training Set and the Conflated Label Development Set.

Morpheme Detector

For the Morpheme Detector, the Conflated Labels Training Set and the Conflated Label Development Set were altered by replacing labels with OpenNLP's native token SPLIT tag - a special tag, used for marking token boundaries during a tokenization-learning task. The two new sets will be referred to as Detector Training Set and Detector Development Set.

Part of Speech Tagger

For the POS taggers, the 3,026 sentences from Ukwabelana were used. The sentences were automatically randomized, after which 60% were designated to be a POS Training Set, and 40% a test set.

5 Workflow Description

The pipeline for POS tagging a text in this project consists of the consecutive application of several models on a corpus. The corpus to which we apply these models is the CTexT corpus of Zulu. The workflow itself involves three stages: Stage 1 is preprocessing steps, Stage 2 is morphological analysis, and Stage 3 is POS tagging. Stages 1 and 2 will be covered in this chapter, while stage 3 will be discussed in detail in a chapter 6.

5.1 Preprocessing Stage

The CTexT corpus comes as a run-on text that in the given form cannot be simply plugged into a pipeline. To be used, it needs to be converted to a format that is suitable for the various NLP tools. This process of convertion is known as preprocessing and involves identifying sentence boundaries (Sentence Detection), and word forms or tokens (Tokenization). In addition, some further processing must be done, so that the data is as similar in format as the data on which the models were trained. In our case that meant removing capitalisation (even though capitalisation provides extra context in languages written with the Latin alphabet) and if necessary conflating or removing punctuation and End-of-Sentence and Start-of-Sentence markers (henceforth: EOS and SOS markers).

5.1.1 Sentence Detection

Sentence Detection takes raw text input and determines sentence boundaries on the basis of a trained model. The output of the Sentence detection task is one sentence per line. In our case that also includes sentence segments such as the text chunks held in the bullets of a bullet list, between semicolons, or a semicolon and an end of sentence punctuation mark.

A model for sentence detection is one of the simplest models to train with the OpenNLP toolkit. For languages written like English, usually only several dozens of sentences suffice to train a reliable sentence detector. As the writing system of Zulu was designed by European missionaries who followed the linguistic traditions of their languages - Latin alphabet, capitalized start of sentences, end-of-sentence punctuation - it was to be expected that the issues should be the same and that sentence detection models for English and Zulu should perform similarly.

While the model is not demanding, the training data still needs to conform to a certain format: to have once sentence per line and to include both positive and negative examples of sentence boundaries, so that the learner can disambiguate. As the Ukwabelana corpus has been stripped of both capitalisation and punctuation, it was unsuitable for training in this form. In order to avoid using the CTexT data, we had to work around this problem in a different manner. During this development phase, several models were built with the following data:

Simple Wiki corpus

We used the manually collected Wikipedia corpus with original Zulu capitalisation and punctuation. It consist of 53 lines with both positive and negative examples of sentence boundaries. After trying out several parameter configurations, we ended up with a model trained for 500 iterations with a cutoff for the features set on 2 (i.e., a feature needs to be seen at least twice in order to be considered in the training). Due to the discrepancies in punctuation between the government website and Wikipedia (an issue of domain and style), some possible EOS markers such as the question mark (?) were missing in the training data. When applied to the CTexT corpus, this model recognised 2,066 sentences.

Augmented Wiki corpus

We expanded the aforementioned Wikipedia corpus with a small set of auxiliary sentences: we repeated ten of the Zulu sentences, but used alternative punctuation to mark the end, in order to provide for EOS punctuation that is only seen in the CTexT data: semicolons and question marks. While these sentences may or may not be grammatically correct, they are formally acceptable, as they allow the model learner (which only looks at the EOS markers) to disambiguate between true and false sentence boundaries. Training with different parameters did not make a difference: with cutoff varying between 5 and 5 and for 100 to 500 iterations, when applied to the CTexT data, though, this model only found 1964 sentences.

Combined Wikipedia and Ukwabelana corpus

The model was trained on the original data from Wikipedia, combined with an auxiliary set: the first 5,500 sentences from the Ukwabelana corpus, with manually added EOS marks - 1,000 question marks, 1,000 semicolons, and 3,500 full stops. After training a model for 500 iterations and no feature cutoff, the CTexT corpus was processed. 2067 sentences were found - one more than with the Simple Wiki corpus.

English model

Finally, the English sentence detection model which comes with the OpenNLP toolkit was applied on the CTexT data. Its performance was completely comparable with both the Simple and the Combined corpus models. It recognised 2,043 sentences.

After examining the output files, we concluded that the models perform comparably, not missing any existing sentences but occasionally splitting unnecessarily when encountering URLs and other non-typical sequences, such as: *607.04.10(03), 607.04.10(09) and 607.04.10(12).* Eventually, the Combined model was used, the justification being that a model which detects more sentences could double up as a chunker, splitting longer sequences and forcing the Maximum Entropy analyser (used later) to consider immediate neighbouring tokens first and foremost.

5.1.2 Tokenization

Once sentences and other large chunks were detected and empty lines removed, the corpus needed to be tokenized. A tokenization model was trained on the Wiki corpus, which was extended with examples of email addresses and URLs from the CTexT data, where they appear infrequently, as well as with two CTexT examples of single quotation marks and ellipsis (...). Finally, two auxiliary sentences containing question marks were randomly selected from the Combined corpus and added to the training data. Examples of loan words with native affixes (signalled with a dash), e.g., *e-Afrika* or *i-website* were left unsplit at this point.

Several models were trained with different cutoff values (0 to 2) and number of iterations (100 to 500). Three-fold cross-validation showed that training with no cutoff for 100 yields the highest scores: Precision: 0.9519, Recall: 0.9387, F-Measure: 0.9452. The folds are limited to three, because of the small amount of training material.

Applying the model on the CTexT corpus, tokenization was carried out correctly, except for isolated cases where an opening parenthesis or an opening single quotation mark was not separated, as well as most forward slashes (e.g., *kwh/kl*). The tokenizer also had trouble with cases where there is a multi-word expression which is also a foreign loan term, hence written with a dash and quotation marks. Such cases were either left unchanged, for example *yi-'duty to maintain'*, or were partially tokenized, e.g., *le-'Save as '*. The corpus itself is not consistent, as such expressions also occur without quotation marks, as is the case with *lwe-Adobe Acrobat Reader 4* or *u-Director-General.* As Taljard and Bosch (2005) point out, the South African Bantu languages as a whole are yet to become more standardised with respect to their orthography and spelling rules (Taljard and Bosch, 2006).

The fact that English expressions would be treated as native words later on was not seen as a problem, since loan words can be incorporated very deeply into the language's system. (Welmers, 1974, p.182) discusses this issue with examples from KeRezi:

> "If the first syllable of the foreign word is sufficiently reminiscent of a prefix in the adopting language, the word is likely to be taken into the class for which that prefix is appropriate, singular or plural, and the pairing prefix is then also used with the reanalyzed stem, as in the case of the pair /i-mato, to-mato/."

Additional preprocessing

The preprocessing was completed by clearing up the rest of the tokenization errors: splitting parentheses, quotation marks, and forward slashes, and by removing capitalisation. Eventually, punctuation that does not mark a sentence boundary was removed: commas, parentheses, quotation marks, bullet points, slashes and sentence-initial whitespace. At this step, dashes occuring within a word form were deleted and the substrings were concatenated. We chose this so that during the next step (morpheme detection and labeling) these word forms are analysed as if they are native Zulu words. Stand-alone English words did not receive special treatment at this point, so they introduce noise. However, the problem of code-switching between different South African languages is a complex one and lies beyond the scope of this thesis. Finally, digits were removed.

5.2 Morphological Analysis

The second stage of the project involves the development of a morphological analyser. This set of models involves a lot of linguistically interesting information: information about the structure of Zulu words, morphological ambiguities, and fuzzy morpheme boundaries. There is a reason why morphological analysis makes sense as a first step. Because Zulu is an agglutinating language which is also written conjunctively, it is often the case that a corpus would feature a large number of long, morphologically complex words, that only occur once: a weak type token relationship. However, "the output of a morphological analyser is a rich source of significant information that facilitates the identification of word classes"(Taljard and Bosch, 2006, p.125).

A large part of the computational work on Zulu involves hand-crafting complex finite-state models of the Zulu morphology and morphotactics (cf. S. Bosch). Alternatively, De Pauw et al. (2012) suggest that linguistically motivated morphological analysis may be a step in POS tagging that can be avoided. Instead, they suggest a pseudo-linguistic word segmentation approach they dub the "Bag of Substrings" approach.

Often in a Zulu word, there is a morpheme that gives enough information on the word category (Spiegler et al., 2010b). It is then reasonable to perform morphological analysis first. The Ukwabelana corpus includes a morphologically analysed word list, which is suitable for training a Morpheme Detector (and subsequently, a Morpheme Labeler). The analyser consists of two models: a model for morpheme detection, which takes a word form as input and outputs a segmented version of it, and a morpheme labeler, which takes the segmented word form and applies a morphological label to each morpheme.

5.2.1 Morpheme Detector

In order to achieve morpheme detection, we took a morphologically analysed word list (one word per line) and treated each word as a sentence, similar to Hakkani-Tür et al. (2002). The Morpheme Detector was trained with a cutoff of 2 for 300 iterations. After ten-fold cross-validation with the OpenNLP cross-validation tool, the reported precision is 0.8156 and the recall 0.8094, with the F-Measure being 0.8127 (81.27%).

False negatives		False positives	
504	a	656	a
274	i	246	e
262	e	128	el
161	o	116	i
135	b	113	n
134	el	101	o
114	y	95	w
113	s	91	ba

Table 5.1: Ambiguous morphemes in Zulu.

In order to understand the issues of the language at hand, we analysed the morphemes which the Morpheme Detector misclassified. We found that there is small number of morphemes which are highly ambiguous, appearing both within the numbers of the false negatives and the false positives. These morphemes can be seen in table 5.1.

What one notices first is that almost all of these morphemes are vowels. This is not surprising, because two of the characteristics of Zulu are that nouns have an initial vowel (cf. table 2.1), and that verbs end in a vowel (often -*a*). The negative prefix for verbs is also *a*-, and in general morphemes in Zulu exhibit vowel harmony, so one expects to see morphemes whose allomorphs are vowels.

These ambiguities may present a problem for the Morpheme Labeler, which has to associate these morphemes with a large number of tags. A big enough corpus would alleviate the problem by providing sufficient data for the learner to infer relationships. For a small corpus like this, though, one would need to find alternative approaches. Our solution was to attempt to reduce the tagset itself. The following section addresses this issue in greater detail.

5.2.2 Morpheme Labeling

POS Tagger as Morpheme Labeler

As a POS tagger functions as a sequence labeler, assigning tags to a sequence of tokens, it can be used to label morphemes as well, provided that there is training data. The morphologically analysed word list that was used in the training of the Morpheme Detector served as the basis for developing training data for the labeler as well.

Original Labeling Scheme

At this point a decision had to be made: to use the original label set of the Ukwabelana Corpus or to design a new one. The argument in favour of the original label set is that it has been tailored to the specifics of Zulu. It was "[...] based on the idea that each morpheme in a word should be labeled, even when words belong to a very restricted class" (Spiegler et al., 2010a). The resulting scheme consists of 206 morpheme labels. The original set thus carries an immense amount of linguistic information. This means that a dedicated language technologist can design many feature sets, ranging from very specific to very abstract ones, and use them to augment or bias higher-level applications (e.g., a POS tagger, a syntactic parser, etc.) in various ways.

The drawback of the original labeling scheme is the large number of labels. While this may be necessary for rule-based applications which benefit from fine-grained distinctions, given the small dataset we have, we feared that the large label set would exacerbate the problem of data sparsity: the ambiguous morphemes that the Detector has found would be associated with a large number of tags, but there would not be sufficient data for the learner to learn to disambiguate between the different cases. In addition, a small number of tokens and a large tagset would result in a large number of singleton pairs. Again, this would mean that subsequently, the POS tagger would not be able to infer the necessary relationships from the morphological data and would fail to associate certain morphological relations or sequences with the respective parts of speech.

Conflated Labeling Scheme

"The degree of granularity of a tagset should be appropriate to the purposes of the tagged corpus"(Allwood et al., 2003)[1]. In accordance with that idea, we folded the original label set of the Ukwabelana corpus to 66 new labels. The labels that were affected are primarily the affixes for the noun classes and their object concords: where the Ukwabelana label set has a separate label for each affix, we only kept a general label for the affix function. For example, where the original label set had 12 labels for demonstrative agreement (denoted *d1* to *d15*), we only have a single label *da* for demonstrative agreement.

[1] Paraphrase quoted from Taljard and Bosch (2006)

Original Scheme		Conflated Scheme	
Type	Error count	Type	Error count
adv	63	adv	66
ar	31	ar	36
gClass	53	g	33
iClass	146	i	71
iv	25	iv	22
iv-nClass	43		
locpf	21	locpf	16
n	65	n	46
nr	72	nr	77
oClass	94	o	56
pClass	97	pp	56
sClass	79	s	44
vr	33	vr	35
zClass	72	za	28
TOTAL Errors	1,209	TOTAL Errors	856

Table 5.2: Mislabeled types. Note: *Class refers to any Zulu noun class 1 to 15

Training, Cross-Validation, and Error Analysis

We trained two Labeler models - one with each labeling scheme - with a cutoff of 2 for 300 iterations, as these parameters gave the best accuracy scores in ten-fold cross-validation. The estimated accuracy immediately showed that a coarser-grained label set performed better than a finer-grained one. The accuracy of the Original Scheme model is 82.55%. For the Conflated Scheme model, the accuracy climbed up to 88.48%. These models were trained with the OpenNLP POS Tagger learner with in-built features, so we did not have any control over the feature selection.

The wrong predictions we compared in terms of errors per token and errors per type. The Original Scheme model labeled 11,437 tokens incorrectly, while the Conflated Scheme model mislabeled only 7,548 tokens, which is a vast improvement. Many of the mislabeled morphemes are the ones which were also problematic for the Morpheme Detector.

The types of errors that occur are represented by the pairs of reference tags vs. predicted tags. Again, where the Original Scheme labeler makes 1,209 errors, the Conflated Scheme labeler only makes 856. Some type of errors were slightly increased, such as those the the label *ar* and *nr*. However, for other types the errors was much lower, for instance for the iClass (146), which dropped by half to 71, for n (from 65 to 46), and for z (from 72 to 28). Some more examples can be seen in table 5.2.

6 Part-Of-Speech Tagger

This section presents the range of experiments on the final work bundle in the current project - the training of a custom part-of-speech tagger for Zulu. Two ideas were of vital importance for this: a negative and a positive view. On the negative point of view, our work was motivated by the intuition that a POS tagger which bases its feature generation of European languages could not handle the complex relationships inherent to Bantu morphology, especially in a low-resource situation. On the positive side, work by De Pawl and de Schryver on resource- and knowledge-light approaches to POS-tagging for African languages showed that with an appropriate machine learning algorithm, it is possible to build a POS-tagger which relies on random orthographic substrings.

Therefore, we decided to explore further De Pauw's and de Schryver's concept of what they call the "Bag of Substrings", and attempt to build a tagger which takes linguistically-motivated substrings. In other words, we wanted to make a tagger which employs the information that our Morpheme Detector provides. Moreover, we wanted to attempt including even more complex features, which take into account not only which substrings carry linguistic information (i.e. are actual morphemes), but also what kind of information is encoded (i.e. what morphological label has been assigned to each morpheme by the Morpheme Labeler).

In order to investigate the different scenarios, we trained three baseline systems and a number of augmented systems with different feature configurations. All the systems were trained on the same training material from the Ukwabelana corpus: 60% of the POS-tagged sentences (12,595 tokens), a random selection. They were all tested on the remaining 40% (8821 tokens). The test consisted of applying the tagger on an unlabeled variant of the test data, and then comparing the predicted tag for each token to the original tag of the token. Thus, the 40% of the data served as a Gold Standard, and this is how we are going to refer to the list of token and tag pairs from this data split.

The tagset consists of 16 tags/outcomes. All systems were trained for 500 iterations and no cutoff as initial trial experiments showed that the final results are not influenced, although the probabilities of individual tags vary a little. In order to keep track of all systems, we adopted a naming convention, where each system is named alphabetically after the NATO (military) alphabet: Alfa, Bravo, Charlie, etc. When within a system several models were trained with different configurations, these were named alphabetically with human names (Annie, Bobbie, Carrie, etc). Thus, for example, system Delta has the submodels Delta-Annie, Delta-Bobbie, and so on. The different feature configurations are in table 6.1

6.1 Baseline Systems

6.1.1 OpenNLP POS Tagger with Features for English (Alfa)

Experiment setup The first baseline system is for the OpenNLP tagger with fixed features for Indo-European languages. The output of the tagger was compared agaist the Gold Standard (the same tokens, with the tags that they came with from the Ukwabelana corpus).

Results Out of 8,821 tokens, 2,066 were matched correctly. The large majority (6,755), however, were mismatched. The total accuracy for this tagger is 23.42%.

6.1.2 MaxEnt Classifier for Unigrams (Bravo)

Experiment setup This baseline aimed to asses how well the tagger can predict when all the context it sees is the token itself.

Results Surprisingly, this system performed much better than the OpenNLP POS tagger, with 69.50% accuracy, calculated in the same way. 6,131 tags were correctly matched, while 2,690 were mismatched.

6.1.3 MaxEnt Classifier for Trigrams (Charlie)

Experiment Setup This system takes as features the token and its left and right neighbour. For this experiment we added with end of sentence and start of sentence tokens, but in later n-gram experiments we discarded this information.

Model	Token	Feature	Tag
Alfa	ngisho nje ...	ngisho_v nje_adv	
Bravo	ngisho	ngisho	v
	nje	nje	adv
Charlie	ngisho	Prev=sos Current=ngisho Next=nje	v
	nje	Prev=ngisho Current=nje Next=ukuba	adv
Delta-Annie	ngisho	ngi sho	v
	nje	nje	adv
Delta-Bobbie	ngisho	Start=ngi Morph=sho	v
	ukuba	Start=u Morph=ku Morph=ba	conj
Delta-Carrie	ngisho	Start=ngi End=sho	v
	ukuba	Start=u End=ba	conj
Delta-Eddie	ukuba	Start=u Morph=ku End=ba	conj
	abonge	Start=a Morph=bong End=e	v
Echo	ngisho	Start=ngi Second=sho End=sho Next=nje	v
	nje	Start=nje End=nje Previous=ngisho Next=ukuba	adv
	ukuba	Start=u Second=ku End=ba Previous=nje Next=abonge	conj
	ukuba	Start=u Second=ku End=ba Previous=nje Next=abonge	conj
Foxtrot-Annie	ngisho	Label=ngi_i Label=sho_vr Start=ngi Morph=sho	v
	nje	Label=nje_ar Start=nje	adv
Foxtrot-Bobbie	ngisho	Label=ngi_i Label=sho_vr Start=ngi Morph=sho	v
	nje	Previous=ngisho Label=nje_ar Start=nje	adv
Foxtrot-Carrie	ngisho	Next=nje Label=ngi_i Label=sho_vr Start=ngi Morph=sho	v
	nje	Next=ukuba Label=nje_ar Start=nje	adv

Table 6.1: Feature combinations for each model.

Results With trigrams, the accuracy jumped by over 10% in comparison with the unigram system. With an accuracy of 80.98%, 7,143 of the tags matched the Gold Standard tags, and 1,678 were mismatched.

6.2 Augmented Systems

These systems build upon the baseline systems in that they include additional morphological information. The first set of systems experiment with adding morphemes and some rudimentary information about their position in the word. The second set expands this idea by combining morphological information to token bigrams. The third set also attempts to improve the tagger with including labels for each morpheme.

6.2.1 Unigrams with Morpheme Detection (Delta)

Delta-Annie

Experiment Setup This is the first of the experiments with the linguistically-motivated output of the Morpheme Detector as additional context. The input features give the token first, then the morphemes and the tag for the token without encoding any additional information.

```
ngisho ngi sho v nje nje adv
```

Results Already, there is an increase in the accuracy to 88.73%. 7827 of the tags are correctly matched and 994 are wrong.

Delta-Bobbie

Experimental Setup This is the same sequence of unigram and morphemes, but some additional information is passed by distinguishing the first morpheme from the token. The intuition behind this feature is that some prefixes that come in the leftmost position would be indicative of certain parts of speech, for instance the prefix for negation - of verbal forms.

```
ngisho Start=ngi Morph=sho v ukuba Start=u Morph=ku Morph=ba conj
```

Results The accuracy rises by 2% to 90.96%, which we could consider an indication that at least some connections are made. 8,024 are correctly matched tags and 797 mismatched.

Delta-Carrie

Experiment Setup This purpose of this feature configuration is to establish if some part of the word segmentation can be discarded. The feature consists only of the leftmost and rightmost morphemes.

```
ngisho Start=ngi End=sho v nje Start=nje End=nje adv ukuba Start=u
End=ba conj
```

Results The accuracy drops by one percent but still this feature configuration remains above the baselines. The tagging accuracy is 89.80%, 7,922 tags match the Gold Standard, and 899 are mismatched.

Delta-Eddie

Experiment Setup This feature set combines the information from Delta-Bobbie, -Carrie, and -Donnie: we provide the entire set of morphemes, but also give prominence to start and end morphemes, which could be indicative of concordat prefixes and sufixes.

```
ukuba Start=u Morph=ku End=ba conj abonge Start=a Morph=bong End=e v
```

Results This is the best-performing feature set: the accuracy is 92.31%, the number of tags that match is 8,143, while for the mismatches it is 678.

6.2.2 Trigrams with Morpheme Detection (Echo)

Experiment Setup

This feature set diverges from the other sets, because it combines a lot of different information. There is the morphological analysis, prominence of first, secon and last morphemes, as well as previous and next token.

```
ngisho Start=ngi Second=sho End=sho Next=nje v nje Start=nje End=nje
Previous=ngisho Next=ukuba adv ukuba Start=u Second=ku End=ba
Previous=nje Next=abonge conj
```

Results

Given such a detailed set, we feared that the inherent relationships may be introducing more noise to the system. There is a drop in accuracy (91.13%), but the system is still among the top performers. There are 8,039 correctly matched tags and 782 mismatched ones.

6.2.3 N-grams with Full Morphological Analysis (Foxtrot)

These systems all include a new element - morpheme label information. The system is incomplete, in that we have not yet experimented with different formats for introducing the morpheme labels. Currently, in every model the feature Label is paired with the combined form *morpheme_label*. As the classifier is very sensitive to input form, it is unclear whether the resulting low scores are due to formatting issues or to problems typical for linguistic data.

Annie

Experiment Setup This system includes the full morphological analysis with labels.

```
ngisho Label=ngi_i Label=sho_vr Start=ngi Morph=sho v nje Label=nje_ar
Start=nje adv ukuba Label=u_iv Label=ku_n Label=ba_o Start=u Morph=ku
Morph=ba conj
```

Results With the accuracy dropping down to 64.80%, this is still the best system of the four. 5,716 tags were correctly matched and 3,105 were mismatched.

Bobbie

Experiment Setup This feature set has features for the full analysis with the first morpheme prominent, and labels for the morphemes.

```
ngisho Label=ngi_i Label=sho_vr Start=ngi Morph=sho v nje
Previous=ngisho Label=nje_ar Start=nje adv
```

Results The accuracy drops to 64.08%. There are 5,653 matched tags and 3,168 unmatched tags.

Carrie

Experiment Setup This set uses a bigram setup, i.e. in addition to the full labeled analysis it also knows the next token.

```
ngisho Next=nje Label=ngi_i Label=sho_vr Start=ngi Morph=sho v nje
Next=ukuba Label=nje_ar Start=nje adv
```

Results Accuracy drops to 63.29%. Tag matches: 5583. Tag mismatches: 3238.

Donnie

Experiment setup The setup is the same as in Carrie, but with trigrams.

```
ngisho Next=nje Label=ngi_i Label=sho_vr Start=ngi Morph=sho v nje
Previous=ngisho Next=ukuba Label=nje_ar Start=nje adv
```

Results Again accuracy drops to 62.79%. Tag matches: 5539. Tag mismatches: 3282.

6.2.4 Comparison of Baseline and Augmented Systems

The classifier performs best when it has information both about word structure, in the form of linguistically-motivated substrings, and of the immediate context. In terms of morphological analysis, fewer but more precise linguistic features are better than many but confusing features. They do not need to be as precise as rules, but basic morphotactics needs to be encoded, such as concord between starting and ending morphemes.

6.2.5 Linguistic Substrings vs. "Bag of Substrings"

While we cannot directly compare our systems with the "Bag of Substrings" system, because of different evaluation metrics, there are many ideas of De Pauw's and de Schryver's that cross with our issues here. First of all, even though they use automatic feature selection and we craft out features, we agree that a knowledge-light approach where substring information is used without morphological labeling (systems Delta and Echo) is a robust and viable approach. Another point of convergence is the form of features: detailed linguistic information may actually be unsuitable for such approaches: the system Foxtrot is a warning against overuse of complex linguistic features. More abstract features about position of salient morphemes in a language fare better (cf. the memory-based learner of De Pauw and de Schryver, and our systems Delta and Echo).

An issue that still remains to be investigated more in-depth is the importance of context. With the memory-based system and the "Bag of Substrings" system context appears to be only of partial importance. With our MaxEnt approach, adding immediate neighbouring token context seems to improve the accuracy, as long as the morphological information is kept on the level of detected morpheme boundaries (i.e., no morpheme labeling).

7 Steps Towards Cross-Lingual Adaptation

The larger context for the current project is to explore the possibilities for cross-lingual adaptation of POS taggers and morphological analyzers between the Bantu languages. Despite the huge number of languages in the group, they are very similar to one another. Still, variations do exist, and languages that belong to the same sub-group share more characteristics to those outside the group.

The language considered for the adaptation is Xhosa, another Bantu language of the Nguni group. Choosing it stands to reason. First, as a Nguni language it is very similar to Zulu, which means that presumably we could pursue adaptation strategies that are easy to control. The fact that there is much code-switching and borrowing between the two would be to our advantage. Second, over seven million South Africans speak Xhosa, meaning that there is a very large community which would benefit from a wider availability of language technology applications in their tongue. Third, as work on the adaptation of morphological analysers for Nguni languages has already been under way in the rule-based paradigm (Bosch et al., 2008a), it is reasonable to contribute to the body of research by exploring an alternative paradigm.

Unfortunately, the inherent limitation of working with low-resource languages prevented us from completing this stage of the research. While we obtained an unanotated Xhosa corpus with the package of parallel corpora by CTexT, acquiring morphologically analysed or POS-tagged data for it was largely unsuccessful despite numerous inquiries. This meant that although we could apply the preprocessing models for sentence detection and tokenization on the unlabeled Xhosa corpus, and be reasonably certain that the output is comparable to the one for Zulu, we could not evaluate the output either of a morphological analyser or of a POS tagger.

7.1 Possible Adaptation Strategies

7.1.1 Direct Application

Nevertheless, we could still conceive of a number of ways to approach the problem of tool adaptation. In this simplest approach, no models would be built specifically for the target language. Provided that the Zulu processing system is sufficiently refined, the models could be applied on the Xhosa data directly. The morphological analyzer (morpheme detector and labeler) would function as a black box and its output would be fed into the POS-tagging system without further evaluation. Only the POS-tagged output would need to be evaluated.

7.1.2 Domain Adaptation Through Self-Training

In addition, the fact that we have parallel data for all South African languages means that we could use them for domain adaptation. The key is the fact that the training material for the models comes from biblical and fiction texts, while the corpus on which it is applied (the CTexT corpus) is from public domain texts. The procedure for the domain adaptation would involve self-training: the morphological analyser would be applied to a portion of the CTexT Zulu data, after which the sequences with the highest probabilities would be appended to the training data and the model would be trained again. This cycle would be repeated for several iterations. Finally, the model would be applied on the Xhosa data from the same corpus for testing.

7.1.3 Feature Biasing

Yet another strategy for the adaptation could to enhance the Zulu POS model with some features that would bias the tagger in favour of morphological analyses that are more likely to occur in the target languages. One could even conceive of a rudimentary rule-based filter that could be applied at an early stage of the workflow in order to cope with specific cases. Finally, these strategies could be combined with any of the other strategies described earlier in this section.

8 Conclusion and Future Work

8.1 Conclusion

This thesis contributes to the body of research on NLP for African languages by investigating whether an accurate part-of-speech tagger for a low-resource African language can be improved by augmenting sparse annotated data with some automatically generated morphological information. For the purpose of this, we trained a Morpheme Detector and a Morphological Labeler for Zulu on the basis of the OpenNLP implementation of the Maximum Entropy framework. Subsequently, we trained a large number of systems with both generic features for Indo-European languages and with custom configurations of features, abstracted from our knowledge about the morphology of Zulu and the Bantu languages in general.

From the experiments we confirmed that standard features for Indo-European languages may not be directly applicable to typologically different languages such as Zulu and the Bantu. Also, our systems indicated that while some linguistic information such as morpheme boundaries and number and position of morphemes may perform better in comparison with simple n-gram approaches, encumbering the system with information about morpheme function (i.e., with labels) may in fact impair the performance.

8.2 Future Work

8.2.1 Further Feature Configurations

The output of the MaxEnt classifier with custom features suggests that these feature sets can be refined further. In particular, the configurations for the Foxtrot system (n-grams and analyzed and labeled morphemes) must be modified to a format that the classifier can interpret. Only then can we understand better if the decreased accuracy scores resulted from a technical problem with the format or if it an issue with the approach itself: it is well known that introducing too much linguistic information into probabilistic systems often causes performance to drop.

Subsequent feature tuning must also be combined with cross-validation of the models, so that the systems can be evaluated against the comparable ”Bag of Substrings” system. Such validation would also give us a measure of how well the models generalize.

8.2.2 Adaptation for Related Languages

Once the best-performing Zulu systems have been identified, the next step would be to select a one or a few Bantu languages as candidates to be the target language for adaptation. The two most important strategies to pursue would be the direct application of the source model on the target language, and the domain adaptation through self-training and parallel corpora.

8.2.3 Hidden Markov Models (HMM) POS Tagger

Finally, it would be interesting to attempt to build a similar system, but based of an HMM tagger. The HMM tagger handles tag generation for strings in a way which may prove to be more robust. Again, an off-the-shelf toolkit and API could be used.

Acknowledgements

My heartfelt thanks go to my supervisors Dr. Alexis Palmer, Mr. Mike Rosner, and Prof. Dietrich Klakow for their invaluable support during the writing of this thesis. The evaluation script for the MaxEnt classifier was kindly written by Gideon Kotzé, who also shared his knowledge and perspective with me on many occasions. Thank you, Marc Schulder, for all the help with formatting my files, proofreading and patience in moments of crisis.

Bibliography

J. Allwood, L. Grönqvist, and AP Hendrikse. Developing a tagset and tagger for the African languages of South Africa with special reference to Xhosa. *Southern African linguistics and applied language studies*, 21(4):223–237, 2003.

Chris Biemann. Unsupervised part-of-speech tagging in the large. *Res. Lang. Comput.*, 7(2-4):101–135, December 2009. ISSN 1570-7075. doi: 10.1007/s11168-010-9067-9. URL `http://dx.doi.org/10.1007/s11168-010-9067-9`.

S. Bird and G. Simons. Seven dimensions of portability for language documentation and description. *Language*, pages 557–582, 2003.

S.E. Bosch, L. Pretorius, and J. Jones. Towards machine-readable lexicons for south african bantu languages. *Nordic Journal of African Studies*, 16(2):131–145, 2007.

Sonja Bosch, Laurette Pretorius, Kholisa Podile, and Axel Fleisch. Experimental fast-tracking of morphological analysers for nguni languages. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2588–2595, Marrakech, Morocco, May 2008a. European Language Resources Association (ELRA). ISBN 2-9517408-4-0. http://www.lrec-conf.org/proceedings/lrec2008/.

Sonja E. Bosch, Laurette Pretorius, and Axel Fleisch. Experimental bootstrapping of morphological analysers for nguni languages. *Nordic Journal of African Studies*, 17: 66–88, 2008b.

G.T. Childs. *An introduction to African languages.* John Benjamins Publishing Company, 2003.

Mathias Creutz and Krista Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. In *Helsinki University of Technology*, 2005.

Guy De Pauw and Gilles-Maurice de Schryver. *African Language Technology: the Data-Driven Perspective*, pages 79–96. Europ, Bolzano, Italy, 2009. ISBN 978-88-88906-52-2.

Guy De Pauw, Gilles-Maurice de Schryver, and J. van de Loo. Resource-light Bantu part-of-speech tagging. *Language Technology for Normalisation of Less-Resourced Languages*, page 85, 2012.

Aditi Sharma Grover and Gerhard B. Van Huyssteen. An hlt profile of the official South African languages. In Guy De Pauw, H.J. Groenewald, and Gilles-Maurice de Schryver, editors, *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pages 3–7, Valletta, Malta, 2010. European Language Resources Association. URL `http://aflat.org/files/grover.pdf`.

D.Z. Hakkani-Tür, K. Oflazer, and G. Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410, 2002.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2 edition, 2008. ISBN 0131873210. URL `http://www.amazon.com/Language-Processing-Prentice-Artificial-Intelligence/dp/0131873210%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0131873210`.

K. Koskenniemi. Two-level model for morphological analysis. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 683–685, 1983.

G.G. Lee, J. Cha, and J.H. Lee. Syllable-pattern-based unknown-morpheme segmentation and estimation for hybrid part-of-speech tagging of Korean. *Computational Linguistics*, 28(1):53–70, 2002.

W.D. Lewis, R. Munro, and S. Vogel. Crisis MT: Developing a cookbook for MT in crisis situations. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 501–511. Association for Computational Linguistics, 2011.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proc. of LREC*, 2012.

L. Pretorius and S.E. Bosch. Finite-state computational morphology: An analyzer prototype for zulu. *Machine Translation*, 18(3):195–216, 2003.

Laurette Pretorius and Sonja Bosch. Exploiting cross-linguistic similarities in Zulu and Xhosa computational morphology. In *Proceedings of the First Workshop on Language Technologies for African Languages*, pages 96–103, Athens, Greece, March 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W09-0714`.

B. Roark and R.W. Sproat. *Computational approaches to morphology and syntax.* Oxford University Press, 2007.

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. Ukwabelana - an open-source morphological Zulu corpus. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1020–1028, Beijing, China, August 2010a. Coling 2010 Organizing Committee. URL `http://www.aclweb.org/anthology/C10-1115`.

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. Additional material for the Ukwabelana Zulu corpus. Technical Report CSTR-10-003, University of Bristol, July 2010b. URL `http://www.cs.bris.ac.uk/Publications/Papers/2001225.pdf`.

O. Streiter, K.P. Scannell, and M. Stuflesser. Implementing NLP projects for noncentral languages: instructions for funding bodies, strategies for developers. *Machine Translation*, 20(4):267–289, 2006.

Elsabe Taljard and Sonja E. Bosch. A comparison of approaches to word class tagging: Distinctively versus conjunctively written Bantu languages. In Isabella Ties, editor, *Proceedings of the Lesser Used Languages and Computer Linguistics Conference (LULCL 2005)*, pages 117–131, Bolzano, Italy, 2006. EURAC research. ISBN 88-88906-24-X.

W.E. Welmers. *African language structures.* Univ of California Press, 1974.