**Morphological Analysis of Ateso using a Finite State Machine**

**Abstract**

In this paper we present the development of a morphological system for Ateso. We used finite state transducers to model Ateso morphology. Xerox finite state tools were used to build the lexical transducers for the system and to model rewrite rules prevalent in Ateso.

An acceptable representation of Ateso morphology is achieved by the system. Ateso language being highly ambiguous, made it really difficult for us during morphological analysis. Regardless we achieved an accuracy of about 86%.

**1 Introduction**

The first step in many natural language processing tasks, such as parsing, machine translation, information retrieval, and part-of-speech tagging, is morphological analysis. In recent years, there have been substantial developments in the field of natural language processing. Many African languages, however, have lagged behind in this development. This scenario has been influenced by two key elements. First off, many African nations now have English and French as their official languages, which has decreased the popularity and made it economically unviable to invest in indigenous languages. Second, there are frequently little or no digital linguistic resources accessible for African languages. The development of such language resources from scratch is further complicated by scarce financial resources and a lack of political will. There are efforts to morphologically analyze other African languages in Kenya for example, Dholuo, Swahili and Kamba.

**2 Literature Review**

**2.1 Previous work**

The morphology of African languages has been the subject of numerous studies. Morphological analysis has been carried out using a variety of methods. De Pauw and Wagacha, for instance, used unsupervised techniques to discover the morphology of Gikuyu in 2007. To automatically create shallow morphological features for Gikuyu, they used maximum entropy learning. This strategy is intriguing since it requires less human labor, but finding a large enough corpus for learning presents a problem.

The Gikuyu language has been studied in other natural language processing (NLP) investigations. Using a grapheme-based memory model, De Pauw et al. (2007) created a model for automatically correcting diacritics.

Researchers at the University of Helsinki in Finland also conducted a morphological analysis study known as SWATWOL in the case of Swahili, a Bantu language (Hurskainen, 2004). This research used the TWOLC tool from Xerox to build two-level rules using a two-level morphology method based on Koskeniemmi's work from 1983.

**2.2 Ateso Morphology**

The Iteso (or people of Teso) are a Nilotic ethnic group in eastern Uganda and western Kenya. Teso refers to the traditional homeland of the Iteso, and Ateso is their language.

The language is lexically similar to closely related languages such as Turkana and Karamojong.

Ateso is a highly inflectional language with a complex word structure and phonemics. The language is also tonal which is a major source of ambiguity. Ateso nouns can be grouped into two categories namely derived and underived nouns. The membership of a noun to a noun class is determined slightly by its initial characters but is mainly determined by the concord system which it enforces on other parts of speech in a sentence.

Ateso verbs are also affected by consonantal and vowel phonemics. Vowel mutation includes vowel lengthening before prenasalised stops and vowel assimilation when some vowel combinations appear in the neighborhood of each other.

**2.3 Finite State Transducers**

It is a well-liked method of computational morphology because finite state machines can be used to formally describe language vocabulary and morphological rewriting rules. Finite state transducers have been used to successfully describe the two-level formalism. In order to determine if an input text corresponds to a language, finite state networks represent grammar rules.

Morphological generators and analyzers are both modeled using finite state transducers. An FST generator converts a lexical form into a surface form using rules. To convert a surface string into the equivalent lexical form, an FST analyzer evaluates both rules and the lexicon. FSTs are bidirectional, therefore the two processes are frequently antagonistic. Finite

state transducers can also be used to carry out other NLP tasks such as tokenization, part-of-speech tagging among others.

**2.4 Two-level formalism**

The two representations can be connected directly in Koskeniemmi's two-level formalism (1983) without the use of intermediary stages. This formalism creates a model that uses parallel rules to connect the two representations. Instead of acting, these rules operate as conditions, confirming the accuracy of the correspondence between the two representations. The parallel modeling of both creation and analysis processes is made possible by this bidirectional formalism. Finite state transducers make it simple to depict as well.

**2.5 Challenges**

Ateso language is more spoken than written, making it difficult to find written sources for the language.

Another source of ambiguity for Ateso language is that it highly depends on the tone you are using.

It is also difficult to find Ateso speakers with in-depth knowledge of the language due to migration of people.

**3 Methodology**

**3.1 Corpus Dataset Collection**

This work uses a 686-word corpus, collected from a variety of resources. Out of this majority is acquired from various internet sources. A small set is also manually transcribed. It is mainly running test and part of it is held out as test data. We cleaned and interpreted the data manually.

**3.2 Model Training**

The model is trained using logistic regression, a widely used classification algorithm. The training data is split into train and test sets using the train\_test\_split function from scikit-learn. The input features (Teso words) are transformed into numerical vectors using the CountVectorizer class, which converts text into a matrix of token counts. The transformed data is then used to train the logistic regression model.

**3.3 Usage**

To use the model, you can enter a word and check if it is classified as an Iteso word. The input word is transformed using the trained vectorizer, and the model predicts its class. The result is then printed on the console.

Please note that in order to run the code and use the model, you need to have the required libraries installed and provide the path to the Teso Corpus Dataset CSV file.

**3.4 How to Run**

* Clone the repository to your local machine.
* Install the required libraries mentioned in the code (pandas, scikit-learn, matplotlib, seaborn).
* Provide the path to the Teso Corpus Dataset CSV file in the code or alternatively save the csv in the same directory as your .ipynb file.
* Run the script, and it will train the model, evaluate its performance, and prompt you to enter a word for classification.

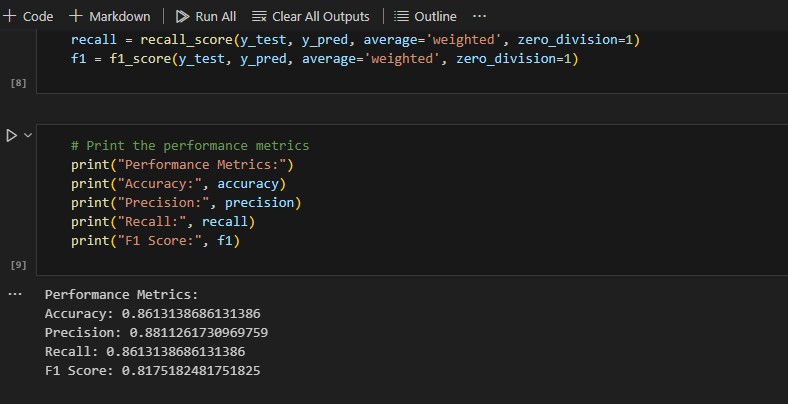
**4 Testing**

In testing we encounter a challenge as Ateso, being a resource scarce language, has no existing standards against which this work can be evaluated against.

After training the model, it is evaluated using various performance metrics such as accuracy, precision, recall, and F1 score. These metrics are calculated using the test set and the predictions made by the model.

Additionally, a confusion matrix is plotted using the confusion\_matrix function from scikit-learn and visualized using the seaborn and matplotlib libraries.

**5 Results**



**6 Conclusion**

We propose that the information and resources gained through this research can be applied to the development of languages that are similar given the close relationship between Ateso and other Nilotic languages.

**Acknowledgments**

The research conducted on this paper was made possible through the support of our fellow student, and under the guidance of our lecturer, Mr. Kituku.

**References**

<https://lughayangu.com/teso>

<https://allthingskenyan.com/countries/kenya/teso-words>

<https://www.cram.com/flashcards/top-1000-ateso-words-420521>

<https://omniglot.com/writing/ateso.htm>