MZUZU UNIVERSITY

FACULTY OF SCIENCE, INNOVATION AND TECHNOLOGY

DEPARTMENT OF INFORMATION, COMMUNICATION AND TECHNOLOGY

CHICHEWA PARTS OF SPEECH TAGGER (CHITAGGER)

SYSTEM PROJECT DOCUMENTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A DEGREE OF BACHELOR OF EDUCATION

IN INFORMATION AND COMMUNICATION TECHNOLOGY

SUBMITTED TO

PROJECT COORDINATOR MR NAMACHA

PROJECT SUPERVISOR MR S. NDEBVU

BY

AUSTIN THAUZENI (BEDICT2918)

11th FEBRUARY 2023

DECLARATION

I Austin Thauzeni (Bedict2918) declare that this project is my original work and has been written by me and has not been submitted for any previous degree. The project work is almost fully my own work, the collaborative contributions have been indicated clearly and acknowledged.

Signature……. Date……..

Supervisor

Mr. S. Ndebvu

Signature……. Date……..

ACKNOWLEGMENT

First of all, I would like thank God who has been with me throughout my college life up to this project and for giving courage to complete this project that shouldn't be taken for granted I adore you for that.

I am greatly humbled by my project supervisor **Mr. S. Ndebvu** for providing valuable guidance, encouragement and inspiration throughout this work.

Special thanks to all my classmates and friends who have been with me and helping me throughout my college life.

Finally, would like to express my heavy gratitude and thanks giving to my dear parents, my uncles and other family members for their financial, moral, spiritual support when I was doing the project.

DEDICATION

This system project has been dedicated to my parents (Mr and Mrs Thauzeni), Mr. and Mrs. Chauluka, Mr. and Mrs. Misheck Chawala, Mr. and Mrs. Maxin Chawala, Mr Chavula and Dr. and Madam Dr Mwale for the financial and moral support that they have showed me throughout my four-year study at Mzuzu University. Finally, not forgetting my brothers and sisters Alinafe Thauzeni, Teleza Thauzeni, Efrida Thauzeni, Miriam Thauzei, Flazer Thauzei, Thandie Chauluka, Mtendere Chauluka, Innocent Chauluka, Luntha Mkwezalamba and Tiyamike Mkwezalamba for their support too

Abstract

Part of speech (POS) tagging is the process of assigning a word in a text as corresponding to a part of speech based on its definition and its relationship with adjacent and related word in a phrase, sentence or paragraph. POS tagging is used as a prerequisite in search engines, Auto spelling completion, named entity recognition, machine translation and other Natural language processing (NLP) applications. The absence of POS tagger affects the efficient retrieval of information in search engines, the accuracy of spell checker in Auto spelling completion, the ability to translate the given text in machine translations and the ability to detect the name of an entity in Named entity recognition. Despite the existence of POS taggers for different languages, Chichewa lacks POS taggers. Therefore, we propose to develop a Chichewa POS tagger to mitigate some of the challenges mentioned above that are faced due to its non-existence.

***Keywords*:** part of speech tagger, NLP, Chichewa

Contents

SECTION 1

CHICHEWA PARTS OF SPEECH TAGGER (CHITAGGER)

SYSTEM PROPOSAL

1.0.Introduction

Language is human tool used for communication. There have been many attempts to simplify the analysis of natural language for various purposes. NLP is meant for such attempt that helps machine to understand the natural language (spoken or written). There are a lot of NLP techniques that can be important as far as NLP applications are concerned. Around these techniques are *Sentiment analysis*: which is the dissection of data (text, voice, etc.) in order to determine whether it is positive, negative or neutral, *text summarization*: which refers to breakdown, tagging and restructuring of text data based on either root system or definition, *tokenization*: it is the breaking down of long-running text string into smaller units called tokens and *stops words removal:* removal of words that do not have any meaning attached to them. Part of speech tagging is one of the concepts of natural language processing. Part of speech tagging refers to categorizing words to a particular part of speech depending on the definition of the words and its context [1]. In recent years, people have been developing parts of speech taggers for different languages. These POS (parts of speech) tools are being used in applications like Named entity recognition, co-reference resolution, speech recognition etc. currently there is lack of parts of speech taggers for Chichewa to be used in some of the mentioned applications. To deal with this problem, this project will provide the Chichewa parts of speech tagger (CHITAGGER). The Chichewa tagger will be used by programmers of natural language processing and researchers who may want to embark on the Natural language processing in Chichewa.

Natural language processing in Chichewa.

1.1. project history

Natural language processing (NLP) is an excellent discipline of computer science focusing on developing artificial intelligence systems that are able to interact with human beings in their natural languages. The expert systems so developed try to understand patterns of human languages and processes the given data (text or speech) accordingly. One of the processes of natural language processing is POS (part of speech) tagging which refers to categorizing words in text (corpus) in correspondence with a particular part of speech depending on the word definition or context. From the past years a lot of part of speech taggers are being developed for different languages. Besides that, there is a lack of POS tagger for Chichewa language. Chichewa

is one of the giant languages spoken in Malawi and some parts of Zambia and Zimbabwe. Like some other languages Chichewa has a lot of ambiguities and it is agglutinative in nature. One word or phrase is a combination of several sub words technically called morphemes. For example, *sindibweraso* (I am not coming again) can be broken as follows *si* (not)-*ndi*(I)-*bwer*(come)-a-*nso*(again). Notice that the "a" has no literal meaning it is just a final vowel to complement *bwer*. Therefore, coming up with a Chichewa POS tagger can be of a great success as it can help to resolve some of the ambiguities and the pre-mentioned agglutinativeness. To add on that, Chichewa tagger can be a necessity to most applications of natural language processing e.g., speech synthesis, speech recognition, machine translation etc.

1.2. literature review

A tagger is a software tool that is used to assign a word to its particular part of speech depending on both the definition of the word as well as the context used [1].A tagger is of the great help to NLP applications which includes speech recognition, information retrieval, collocation, machine translation, frequency analysis and speed synthesis. They have been many recent studies pertaining to part of speech taggers for different languages. Regardless, the effort put for different languages on the parts of speech tagging, no researcher has focused much on Chichewa parts of speech tagger of which brings problems in processing Chichewa text in natural language processing. Some of these problems encountered in search engines whereby Chichewa query fails to give precise results as well as takes a bit longer to be processed. Similar problems might be observed in machine translations, text summarization and many more. This literature review discusses on the work of different methods that are used to do parts of speech tagging including the POS taggers developed with their corresponding results.

[2] developed a tagger which follows supervised machine learning approach and use Conditional Random Field (CRF) which is used to capture patterns of sequences containing code switching to tag each word with accurate part of speech information. Results of this tagger were evaluated by the baseline model (Stanford parser) and CRF model which gives the following results 60.05% on baseline model and 75.22 on CRF model. On Hindi-English it gives 50.87% on baseline model and 73.2% on CRF model and on Tamil-English it gives 61,02% on baseline model and 64.83% on CRF model.

Vingwen et all [3] created a framework for evaluating different methods of part-of-speech (POS). They used a dataset which comprised 10000 sentences from Indonesian news within 29 tags.

The methods of participants ranged from feature-based to neural networks using classical machine learning techniques or ensemble methods. The best performing results achieve accuracy of 95.82% and showing that neural sequence labeling models significantly outperform classic future-based methods and rule-based methods.

Seth et all, [4] construct and evaluated a part of speech tagger for Yiddish. They used 80000-word dataset of the Penn parsed corpus of historical Yiddish. They evaluated their tagger performance on a 10-ford cross-validation split with and without the embeddings, showing the embeddings improve the tagger performance.

On the same note, [5] presented lexical semantics of Sinhala language and developed a POS tagger for Sinhala language using Hidden Markov Model (HMM). Their research based on statistical approach of which tagging process was done by computing the tag sequence probability and word likelihood probability from the given corpus. The results show that the tagger provides more than 90% accuracy of known words.

As seen in [6] purnamasari et all, came up with their rule-based part of speech tagger for Indonesian language. They used Nazief-Andriani algorithm for stemming which gives the highest accuracy. On the part of construction of POS tagger, they used both rule-based and probabilistic and the incorporation of both. They divided the entire process into three main parts namely 1. Tokenization 2. Affix detection 3. POS tagger. After evaluation their tagger produces 87.4% accuracy.

Like others did [7] designed and implemented a parts of speech tagger for Setswana language. They used a rule-based approach to develop morphological analyzer and python NLTK for tagging the individual word. Their results indicated that 300 sentence files give a performance of 74%. The problem is that their tagger fails when it encounters expansion rules not implemented and when tagging by the morphological analyzer is incorrect.

Contrary to other researchers, [8] research outlines on the recently used methods for designing part-of-speech taggers. He came up with approaches based on handwritten local rules, taggers based on ngrams automatically derived from text corpora, taggers based on hidden Markov Models and taggers using automatically generated taggers. According to his results he found out that taggers developed from handwritten local rules perform better than other approaches.

Mashhad I, designed and implemented a Persian language POS tagger which was based on both rule based and statistical based models [9]. This tagger used part of the Bijankhan's tagged corpus which was maintained at the linguistics laboratory of the University of Tehran. The tagger was evaluated with two texts; standard and nonstandard text. Number of tokens that have tagged manual were 2500 and accuracy of the system was 97%. Nonstandard texts are with different topics. Number of topics that have tagged manual are 675 and accuracy of the system is 92%.

Lastly, Mohmand A. [10] did research and develop a system for Arabic Morphological analysis and part of speech tagging called Qutuf. This system became a kernel of a large framework that provides APIs for Arabic language processing. It was designed and implement as a rule based expert system. It also uses an open-source Database for root extraction, pattern matching, morphological feature and POS assignment. The Qutuf used a tagset that was built based on a morphological feature tagset. The output of Qutuf system provide as HTML because was intended to be published and used over the internet and provide xml to be used by other systems.

To sum up, this literature review has discussed different parts of speech taggers developed for different languages. Therefore, this project will follow some of the methods used and try to resolve some of the problems encountered with these taggers.

1.3. problem definition

There are a lot of challenges that NLP applications are experiencing due to the absence of Chichewa parts of speech tagger. Among these problems, include the following

a. Search engines
   inefficiency in retrieving information upon given Chichewa query
b. Auto completion
   Inaccuracy in determine the likely word to complete
c. Named entity recognition
   Inaccuracy in detecting the names of entities upon given Chichewa text
d. Machine translation
   Inability to translate the word from another language (e.g., English) to Chichewa language

Therefore, we intend to design and implement Chichewa parts of speech tagger to reduce some of the aforementioned problems.

1.4. project justification

Chichewa parts of speech tagger (CHITAGGER)

This project upon its successful, Chichewa parts of speech tagger (CHITAGGER) will increase the performance of search engines, Auto completion, named entity recognition, machine translation and other NLP applications. The performance will be in terms of speed, accuracy and efficiency.

1.5. Goals

- To implement Chichewa parts of speech tagger
- To test the accuracy of Chichewa parts of speech tagger

1.6. Specific objectives

- To develop Chichewa tagset
- To design Chichewa parts of speech tagger

2.0. Area under study

The study has designed a tool (system) that can be included/integrated in NLP (natural language processing) applications. The tool has major components which include the tagset (which is the collection of parts of speeches) for a particular language. The application has interface for mere users. On the other hand, programmers can just integrate the tool in their application without the interface. The tool can be installed in the computer for mere users. Users can use the system by entering the text sentence or phrase in the text field. Then the system will be able to process the text and assign a particular part of speech to each word of the text. The system will have the corpus used to train the tagger. The system uses the supervised machine learning algorithms which is based on the both rule based and stochastic approaches of tagging process. On the rule-based approach the tool uses Hidden Markov model (HMM). Rule based approach perform the tagging process using the rules that were developed by language experts while stochastic use probabilities for tagging process.

2.1. Scope

The main aim of this project is to design, develop and implement Chichewa part of speech tagger (Chichewa tagger) that will be able to assigns part of speech tags to the Chichewa text.

My view of this project is to start by studying different approaches of part of speech tagging and use one of the approaches in developing the Chichewa tagger.

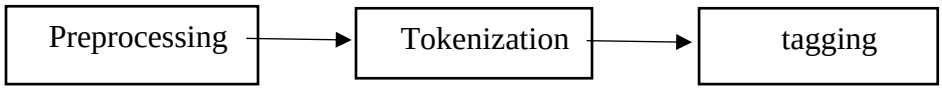
Thereafter, the tagset will be developed which will contains the parts of speech tags to be used to tag a corpus word to its corresponding category of parts of speech. This process will be done by following the available rules and guidelines.

Lastly, the Chichewa tagger will be developed which will be based on the developed tagset and the available corpus (Data set) for Chichewa. There is no need to say that this project will make life easier for researchers, I hope so. Precisely, what this project will provide is going to serve search engines, information extraction applications and any other NLP (Natural Language Processing) that makes use of Chichewa language processing.

2.2. Deliverables

Methodology

This section explains the tagging process with diagram and the use of algorithms (rule based and stochastic based).



Rule based approach: this approach uses contextual and morphological information to assign tags to unknown and ambiguous words. Some of these rules are written by someone or are learned from corpus.

Stochastic approach: this approach uses probabilistic algorithms and need to be trained on a pre-tagged corpus. Exit of tagging part is token and its tag. These are some tags for each token and tags order significantly for each word and stochastic approach select probable tag for word.

Preprocessing stage: on this stage is where the tagger will remove stop words (words that do not have meaning attached to them) as well as removing repeated words. This stage will produce a processed text.

Tokenization: a text produced at the processing stage will be broken into smaller units called tokens.

tagging: the tokens produced will be searched in the corpus in order to find their tags. The results of this stage will be each token plus its tag.

This project will use a python programming language together with some supervised machine learning algorithms and natural language processing concepts.

3.0. project plan

TASK DATE	START DATE	DUE DATE	DURATION

Project title writing and approval			
Proposal documentation			
Proposal presentation			
Proposal first submission			
SRS documentation			
SRS first submission			
SRS presentation			
Detailed design documentation			
Detailed design presentation			
Detailed design submission			
Coding and testing			
Final system presentation			
Final system submission			

3.1. Project budget

ITEM	**QUANTITY**	**COST (MK)**
Internet bundle	50 GB	55,000.00
Pens	2	1,000.00
Ream of papers	1	7,000.00
Pencils	1	5,000.00

Transportation		20,000.00
	TOTAL	88,000.00

4 References

[1] M. Jurasfsky.D, "computational Linguistics and Speech Recognition," vol. 1, no. 2002, p. 20, 2000.

[2] S. Ghosh, S. Ghosh and D. Das, *part-of-speech Tagging of Code-Mixed Social Media Text*, Kolkata: Association for Computation Linguistics, 2016.

[3] F. Shangyi, C. Yingwen, L. Janyi, H. Nankai, Q. Ximan and J. Xinying, *Overview of the Part-of-Speech Tagging Task for Low-resourced Languages*, Guangdong, 2020.

[4] S. Kulik, Nelille, B. Santorini and J. Wallenberg, *A part of speech tagger for Yiddish*, 2022.

[5] A. Jayaweera and N. Dias, "Speech Tagger for Sihnala Language," *International Journal on Natural Language Computing*, vol. 3, no. 2014, p. 12, 2014.

[6] K. Purnamasari and I. Suward, *Rule Based part of speech for Indonesian Language*, IOP publishing, 2018.

[7] G. Malema, B. Tebulo, B. Okgetheng, B. Mothank and M. Rammidi, *Complex Setswana parts of Speech Tagging*, Gaborone, 2020.

[8] A. Voutilainen, *part-of-speech Tagging*, 2012.

[9] I. Mashhad, *A study on the part of speech*, Khavan Higher-Education Institute, 2011.

[10] M. Altabba, A. Zerare and A. Shukary, *An Arabic Morphological Analyzer and part of speech tagger (Qutuf)*, 2010.

5. Appendices

ITEM	MEANING
NLP	Natural Language Processing, refers to the branch of computer science and more specifically the branch of artificial intelligence concerned with giving computers the ability to understand text and speech words in much same way as human being
Tagset	Is a list of part-of-speech tags (POS tags) i.e. labels used to indicate the part of speech and sometimes also other grammatical categories (case, tense, etc.) of each token in a text corpus
Corpus	Is a collection of linguistic data (usually contained in a computer database) used for research, scholarship and teaching
Tagger	A piece of software that adds identifying or classifying tags to pieces of text or data
Python	A general-purpose programming language that can be used on any modern operating system. It be used for processing text, images, scientific data etc.

SECTION 2

CHICHEWA PARTS OF SPEECH TAGGER (CHITAGGER)

SYSTEM REQUIREMENTS SPECIFICATION (SRS)

Abstract

Part of speech (POS) tagging is the process of assigning a word in a text as corresponding to a part of speech based on its definition and its relationship with adjacent and related word in a phrase, sentence or paragraph [1]. POS tagging is used as a prerequisite in search engines, Auto spelling completion, named entity recognition, machine translation and other Natural language processing (NLP) applications. The absence of POS tagger affects the efficient retrieval of information in search engines, the accuracy of spell checker in Auto spelling completion, the ability to translate the given text in machine translations and the ability to detect the name of an entity in Named entity recognition. Despite the existence of POS taggers for different languages, Chichewa lacks POS taggers. This project will come up with Chichewa parts of speech tagger to be used in different natural language processing applications that involves Chichewa language. Chichewa tagger contains these features; preprocessing, tokenization and tagging. The tagger will be implemented following probabilistic/stochastic approach using Hidden Markov Model (HMM) and Viterbi algorithm. It will be developed using python programming language.

***Keywords*:** part of speech tagger, NLP, Chichewa

1 Introduction

Language is human tool used for communication. There have been attempts to simplify the analysis of natural language for various purposes. NLP is meant for such attempt that helps machine to understand the natural language (spoken or written). There are a lot of NLP techniques that can be important as far as NLP applications are concerned. There are three techniques to POS tagging which are *Rule-based approach:* it uses series of rules for tagging process, *stochastic/probabilistic approach:* the tagging process is based on the probability of a word belongs to a particular tag or based on a sequence of preceding/succeeding word. Finally *deep learning approach*: this approach uses deep learning techniques to infer POS tags [2].

Chichewa part of speech tagger will utilize rule-based and stochastic/probabilistic approach for its implementation. Chichewa parts of speech tagger will be implemented using Hidden Markov Model and Viterbi algorithm to increase its accuracy. It will use an annotated Chichewa dataset that has been split into two sets, train-set and test-set. The development of the tool will be divided into the following phases: data collection (this phase utilize the existing dataset), preprocessing (further processing of the raw collected text is required in order to leverage inconsistent writing styles of different contributors), Tag set (is the collection of tags or grammatical classes to which each token in the test dataset has to be classified, tokenization(it is the process whereby raw text is further split into smaller chunks of tokens suitable for further processing, creating custom Chichewa corpus (Chichewa language has no available tagged corpus, therefore we need to create a new one. Specification of feature attributes for HMM and Viterbi algorithm feature functions need to be fed to the model; which is basically a specification of the context of a given word in the sentence and applying HMM and Viterbi algorithm. [3]

1.1 problem statement

Problem statement is a list of challenges which NLP applications are experiencing due to the absence of Chichewa parts of speech tagger. Among these problems include the following:

a. Search engines

   Inefficiency in retrieving information upon given Chichewa query

b. Auto completion

   Inaccuracy in determining the likely word to complete

c. Named entity recognition

   Inaccuracy in detecting the names of entities upon given Chichewa text.

d. Machine translation

   Inability to translate the word from another language (that is from English to Chichewa)

Therefore, the Chichewa parts of speech tagger (CHITAGGER) is being developed to reduce some of these problems.

1.2 project scope

Project scope is part of the project planning and it involves determining and documenting a list of specific project goals, deliverables, features, functions etc. project scope covers what is need to be achieved and the work that must be completed to deliver the project. The project shall perform the following activities

1.2.1 data collection

The project has utilized the already available dataset that has been collected from zindi website

1.2.2 preprocessing

Further processing of the collected raw text from the dataset is required in order to leverage inconsistent writing styles of different contributors [4]. Most of them are a result of grammar in general e.g., inconsistency in some compound words is very common wherein there is some compound word which is written as spaced compound noun, solid compound noun (without any space in between) or as hyphenated compound noun.

1.2.3 Tagset

A tagset is a collection of tags or grammatical classes to which each token in the dataset has to be classified. A tagset for this project comprises of 18 tags

1.2.4 Tokenization

Tokenization is the process whereby raw text is split into smaller chunks of tokens suitable for further processing. This project, the phrases are tokenized into words separated by exactly one space punctuations and symbols re treated as separate words and are thus labeled accordingly.

1.2.5 creation of custom Chichewa corpus

Although, we have the existing dataset, this dataset is not annotated therefore we are required to annotate this dataset so that it should be a tagged corpus. A POS tagged corpus is being created using the web-based application that we have developed.

1.2.6 specification of features

Attributes for Hidden Markov Model and Viterbi algorithm feature functions need to be fed to the models which is basically a specification of the context of a given word in the sentence.

1.2.7 Hidden Markov Model and Viterbi algorithm

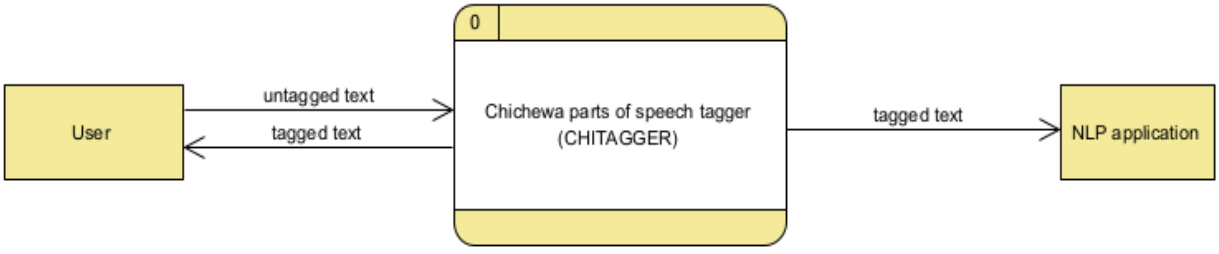
At the end, Chichewa parts of speech tagger will be implemented using supervised machine learning techniques that is by using Hidden Markov model and Viterbi algorithm. These models have been chosen deliberately considering the nature of Chichewa language and the presence of the tagged corpus that we are creating.

1.3 system personnel

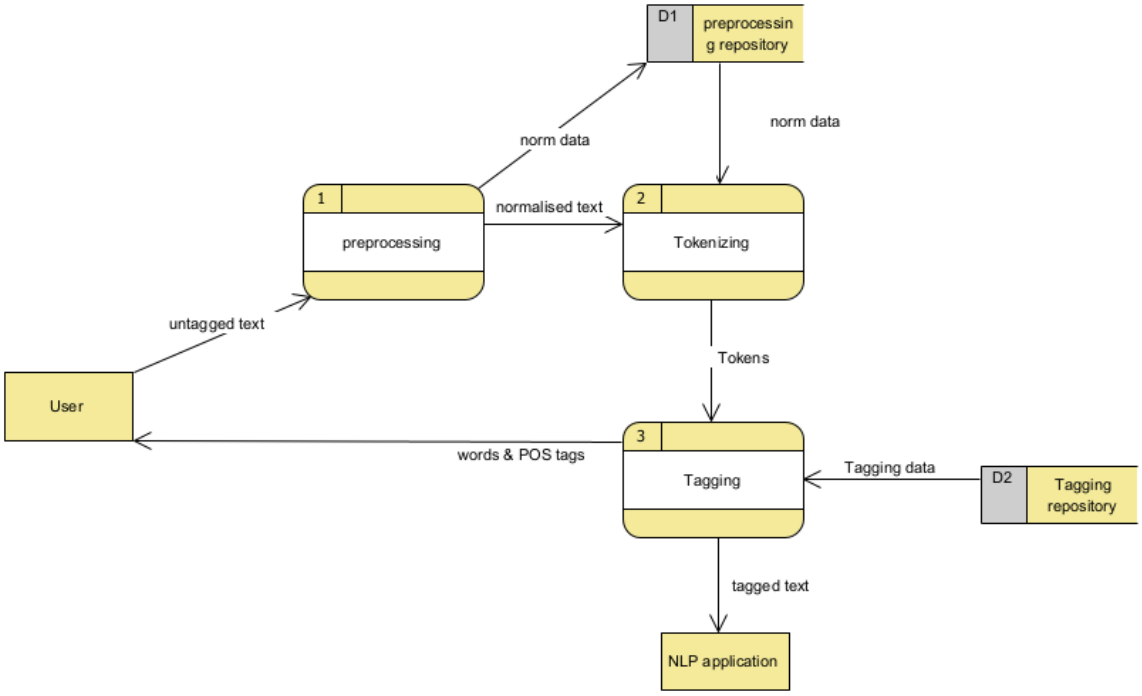
The system will be used by NLP application programmers who may want to develop NLP applications that utilize Chichewa language for example Auto completion. Furthermore, the Chichewa POS tagger will have a graphical user interface for anyone who might want to check parts of speech for a sentence or phrase. The system is being developed by Austin Thauzeni a final year student as a partial fulfilment for the requirement of the award Bachelor of education in information and communication technology at Mzuzu university. The owner of this system is Austin Thauzeni who is also the developer of the system.

1.4 system overview

A context diagram for the proposed diagram



A level 0 diagram for the proposed system



2 System requirements

2.1 Function requirements

Req. number	Requirement	Description
1	The system shall be able to clean text	The Chichewa parts of speech tagger will need to remove punctuations and some unnecessary words from the text before tagging it
2	The system shall be able to tokenize text	Chichewa parts of speech tagger will need to break the given text into tokens in order to assign a part of speech tag to each word

3	The system shall be able to stem the words	The system will need to be able to cut the word into its root from the corpus for easy assigning of a parts of speech tag
4	The system shall be able to assign a part of speech tag	The system will need to assign a part of speech tagger to each word of the text that the user has entered

2.2 Interface requirements

The proposed system shall have a form of interface. The form will be a web based whereby users can provide the query in the text field of the form. The tool shall be able to assign the part of speech of each word of the text and display the results on the same form.

2.3 Non-functional requirements

2.3.1 System-related non-functional requirements

a. Performance

i. Time

Regardless of the network problems, the expected response rate of the system after a user tagging request has been sent, the response of the request should be displayed in less than 3 seconds. The system shall be designed to provide a fast response rate hence making the system more usable

ii. space

The system shall need a device with a minimum RAM of 2 GB for it to run effectively. This can be a personal computer or most mobile devices.

b. Operation environment

i. Hardware platform

The minimum hardware required to support the system is a computer or mobile device with minimum RAM of 2 GB, a processor not less 1.0 GHZ and some free secondary space.

ii. software platform

The system will require a minimum PC operating system such as Microsoft windows 7 and smart phone operating systems for example Android OS, IOS, etc.

c. Standard conformance

The tool will be developed using python programming language and some python Frameworks such as spacy, NLTK, pandas, NumPy, etc. Therefore, the tool shall follow the rules and standards of the mentioned programming language and the rules of the related Frameworks.

d. General characteristics

i. security requirements

anyone can use the Chichewa parts of speech tagger using web-based interface. However, for NLP programmers will need permission from the owner to integrate the tool in their applications

e. Reliability

the tool shall be used anytime and it will be designed in a way that it can improve its efficiency

f. Portability

the tool can be used in any device as far as the device meet minimum requirements of the tool.

2.3.2 User-related non-functional requirements

Skills: since the system will be mostly used by programmers and some mere users. Therefore, programmers will need to have programming skills and knowledge just as it takes for someone to be a programmer. Whereby mere users will require only little computer knowledge

3. project plan

3.1 development cost

The project budget includes the resources together with their amount. The following table lists the resources to be included in the project budget. Most importantly the project will need costs for the people who will be helping in annotating the dataset. Documents will need to be printed whenever they are submitted hence the printing expenses. Internet expenses will be used to access internet by buying bundles to learn things from the internet. Contingency expenses is an extra amount of money to be used when there is budget overrun in the course of the project.

Expenses	**Quantity**	**Amount (MK)**
Printing (proposal, SRS, DDD, user manual)	4	30,000
Airtime	-	10,000

Rim of paper	2	16,000
photocopy	-	6,000
bundles	-	20,000
Pen and pencil	3 pens and 2 pencils	1,000
Contingency expense	-	20,000
Blank disks	4	4,000
annotators	5	50,000
Total		**157,000**

3.2 Software life-cycle constraints

for any successful project, there are still some problems that are encountered along the way. Problems that are to be faced in this project includes, mastering python programming language and some of its frameworks such as NLTK, spacy, pandas, NumPy etc. furthermore, there is a need to learn machine learning and natural language processing models; Hidden Markov Model (HMM), Conditional Random Field and Viterbi algorithm. In addition; the available Chichewa dataset is unannotated. Therefore, we need to annotate the dataset approximately to 100,000 words in order to achieve the best accuracy of the tool. Another problem is concerning with the researching of information and some learning material from the internet.

3.3 system delivery

3.3.1 extent of deliverables

The first deliverable of this project was the proposal document. The second deliverable will be the software requirements specification document. Thereafter, there will be a detailed design document. Lastly, the final deliverable of this project will be the working Chichewa parts of speech tagger delivered together with the user manual.

3.3.2 deliverable format

The deliverable formats of the project documents will be in PDF, DOC and printed documents. The final system will be in executable format in a CD.

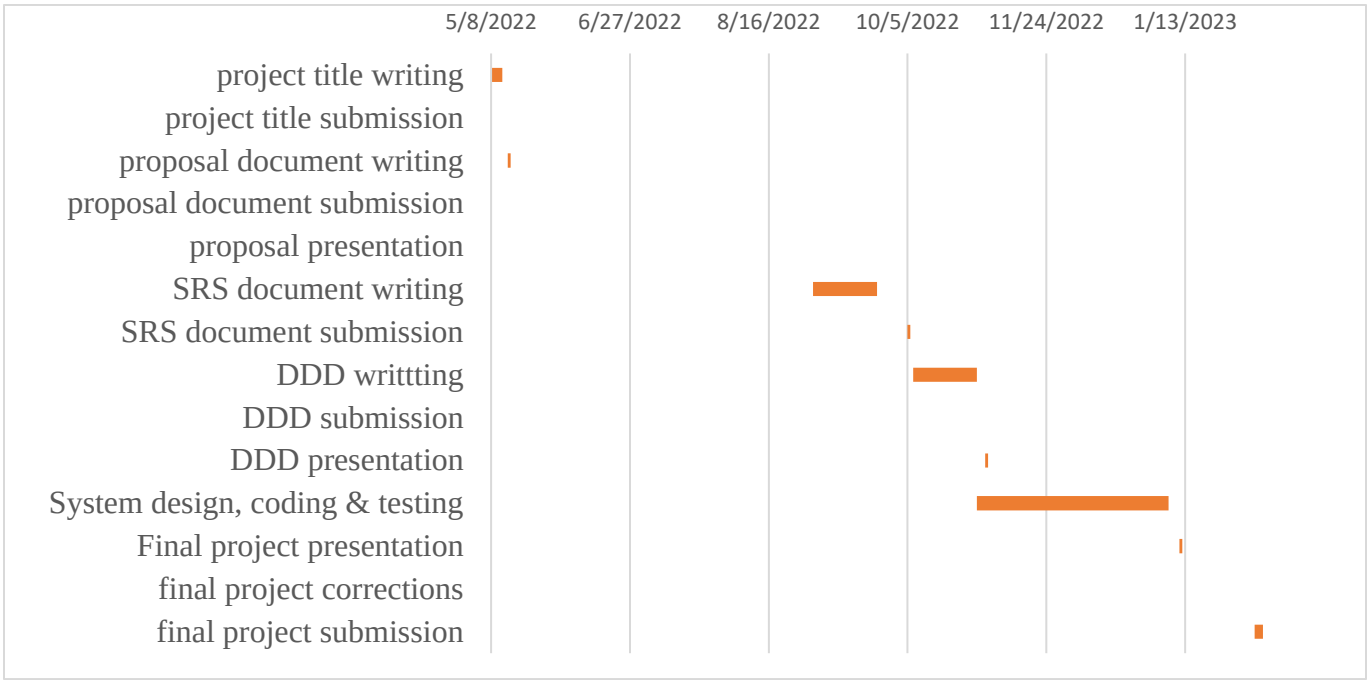
3.4 installation

Upon successful completion of developing the software. The tool will be available online where programmers and other NLP developers can integrate it in their applications. The tool will also have the online graphical user interface where other users can use it when needed.

3.5 development time

The development is shown using the Gantt chart in the following diagram to indicate project plan for the project. The Gantt chart shows the activities together with their durations. The system will take four months from the beginning to implementation. It will have two phases. The first phase shall encompass problem topic identification and writing, proposal writing and presentation and software requirements specification (SRS) document writing. The second phase will include the detailed design document (DDD) writing and presentation. The final system will also be implemented on this second phase together with the user manual. The Gantt chart below outlines the activities to be done during the first phase among with their respective time periods.

Gannt chart showing project activities



References

[1] J. M. D, "computation Linguistic and speech Recognition," pp. 20-100, July 2002.

[2] I. Mashhad, a study on the part of speech, Khavan Higher-Education Institute, 2011.

[3] A. voutilainen, part of speech tagging, New york, 2012.

[4] Z. A. A. Altabba M, "An arabic Morphological Analyser and Part of Speech tagger," Mumbai, 2010.

SECTION 3

CHICHEWA PARTS OF SPEECH TAGGER (CHITAGGER)

DETAILED DESIGN DOCUMENTATION (DDD)

Abstract

Part of speech (POS) tagging is the process of assigning a word in a text as corresponding to a part of speech based on its definition and its relationship with adjacent and related word in a phrase, sentence or paragraph [1]. POS tagging is used as a prerequisite in search engines, Auto spelling completion, named entity recognition, machine translation and other Natural language processing (NLP) applications. The absence of POS tagger affects the efficient retrieval of information in search engines, the accuracy of spell checker in Auto spelling completion, the ability to translate the given text in machine translations and the ability to detect the name of an entity in Named entity recognition. Despite the existence of POS taggers for different languages, Chichewa lacks POS taggers. This project will come up with Chichewa parts of speech tagger to be used in different natural language processing applications that involves Chichewa language. Chichewa tagger contains these features; preprocessing, tokenization and tagging. The tagger will be implemented following probabilistic/stochastic approach using Hidden Markov Model (HMM) and Viterbi algorithm. It will be developed using python programming language.

1. Introduction

1.0 Background

Language is human tool used for communication. There have been attempts to simplify the analysis of natural language for various purposes. NLP is meant for such attempt that helps machine to understand the natural language (spoken or written). There are a lot of NLP techniques that can be important as far as NLP applications are concerned. There are three techniques to POS tagging which are *Rule-based approach:* it uses series of rules for tagging process, *stochastic/probabilistic approach:* the tagging process is based on the probability of a word belongs to a particular tag or based on a sequence of preceding/succeeding word. Finally *deep learning approach*: this approach uses deep learning techniques to infer POS tags [2].

Chichewa parts of speech tagger will utilize rule-based and stochastic/probabilistic approach for its implementation. Chichewa parts of speech tagger has been implemented using Hidden Markov Model and Viterbi algorithm to increase its accuracy. It uses an annotated Chichewa dataset that has been split into two sets, train-set and test-set. The development of the tool has been divided into the following phases: data collection (this phase utilizes the existing dataset), preprocessing (further processing of the raw collected text is required in order to leverage inconsistent writing styles of different contributors), Tag set (is the collection of tags or grammatical classes to which each token in the test dataset has to be classified, tokenization(it is the process whereby raw text is further split into smaller chunks of tokens suitable for further processing, creating custom Chichewa corpus (Chichewa language has no available tagged corpus, therefore we need to create a new one). Specification of feature attributes for HMM and Viterbi algorithm feature functions need to be fed to the model; which is basically a specification of the context of a given word in the sentence and applying HMM and Viterbi algorithm. [3]

1.1 Objectives/goals

The general purpose of the project is to solve the problems that are being faced by Natural Language Processing (NLP) applications due to absence of Chichewa parts of speech tagger. The project shall be able to achieve the following;

- To develop Chichewa tagset
- To design Chichewa parts of speech tagger
- To implement Chichewa parts of speech tagger

- To evaluate Chichewa parts of speech tagger

1.2 scope of solution

project scope is part of the project planning and it involves determining and documenting a list of specific project goals, deliverables, features, functions etc. project scope covers what is need to be achieved and the work that must be completed to deliver the project. The project shall perform the following activities;

1.2.1 data collection

The project has utilized the already available dataset that has been collected from Zindi website

1.2.2 preprocessing

Further processing of the collected raw text from the dataset is required in order to leverage inconsistent writing styles of different contributors [4]. Most of them are a result of grammar in general e.g., inconsistency in some compound words is very common wherein there is some compound word which is written as spaced compound noun, solid compound noun (without any space in between) or as hyphenated compound noun.

1.2.3 Tagset

A tagset is a collection of tags or grammatical classes to which each token in the dataset has to be classified. A tagset for this project resembles that of universal dependency tagset that contains 18 tags. This has been done deliberately to conform to the standards of tagsets.

1.2.4 Tokenization

Tokenization is the process whereby raw text is split into smaller chunks of tokens suitable for further processing. This project, the phrases are tokenized into words separated by exactly one space punctuations and symbols re treated as separate words and are thus labeled accordingly.

1.2.5 Creation of custom Chichewa corpus

Although, we have the existing dataset, this dataset is not annotated therefore we are required to annotate this dataset so that it should be a tagged corpus. A POS tagged corpus is being created using the web-based application that we have developed.

1.2.6 specification of features

Attributes for Hidden Markov Model and Viterbi algorithm feature functions need to be fed to the models which is basically a specification of the context of a given word in the sentence.

1.2.7 Hidden Markov Model and Viterbi algorithm

At the end, Chichewa parts of speech tagger will be implemented using supervised machine learning techniques that is by using Hidden Markov model and Viterbi algorithm. These models have been chosen deliberately considering the nature of Chichewa language and the presence of the tagged corpus that we are creating.

1.3 Constraints

These are some of the factors that can hinder the fast and successful implementation of the full functionality of the Chichewa parts of speech tagger;

- The resources to use such as Chichewa grammar are scarce
- Unable to find people to assist in annotating the Chichewa dataset
- Unavailability of enough Chichewa syntactic rules to help in improving the accuracy of the system
- Difficult in balancing time between working on the project and time for other courses as well

2.0 Architectural and component-Level design

This section provides description of the system architecture with the relationships of all the system components, it describes the architectural diagram of the system which are the data flow diagrams of Level 0, Level 1 and Level 2. The section further presents the description of the components, the detailed description of the constituent processes of each component.

Component description

User module: This contains all the operations needed for the users who will be annotating and reviewing the dataset. It contains functionalities such as Login, register and operations concerning with authentication etc.

Preprocessing: this module deals with preparation of the text before start operating on it. Tasks of preprocessing includes cleaning, removing stop words etc.

Tokenization: in this module is where the prepared text is split into small units called tokens depending on white space and punctuation.

Tagging: on this module is where important work is done. This is where training of the model occurs. Hidden Markov model and Viterbi algorithm are implemented in this module. The module also contains some features of Chichewa language that helps in tagging unknown words to achieve a good accuracy.

2.1 Architectural diagram

This is the pictorial representation of the system and its constituent components using Data Flow Diagrams.

2.1.1 context diagram of the proposed system

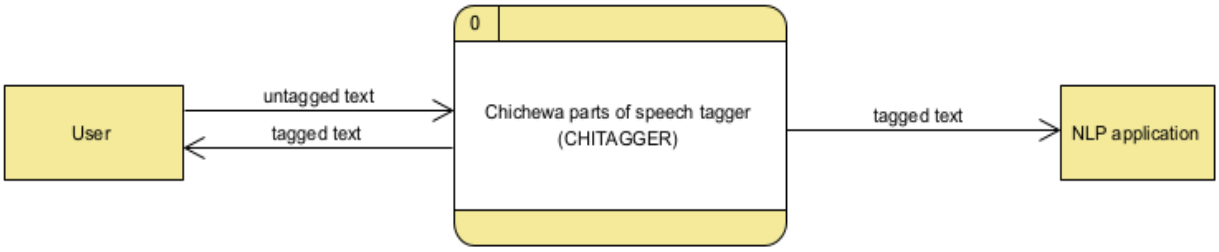


Figure 1: context diagram for Chichewa parts of speech tagger

2.1.2 Level 1 diagram of the proposed system

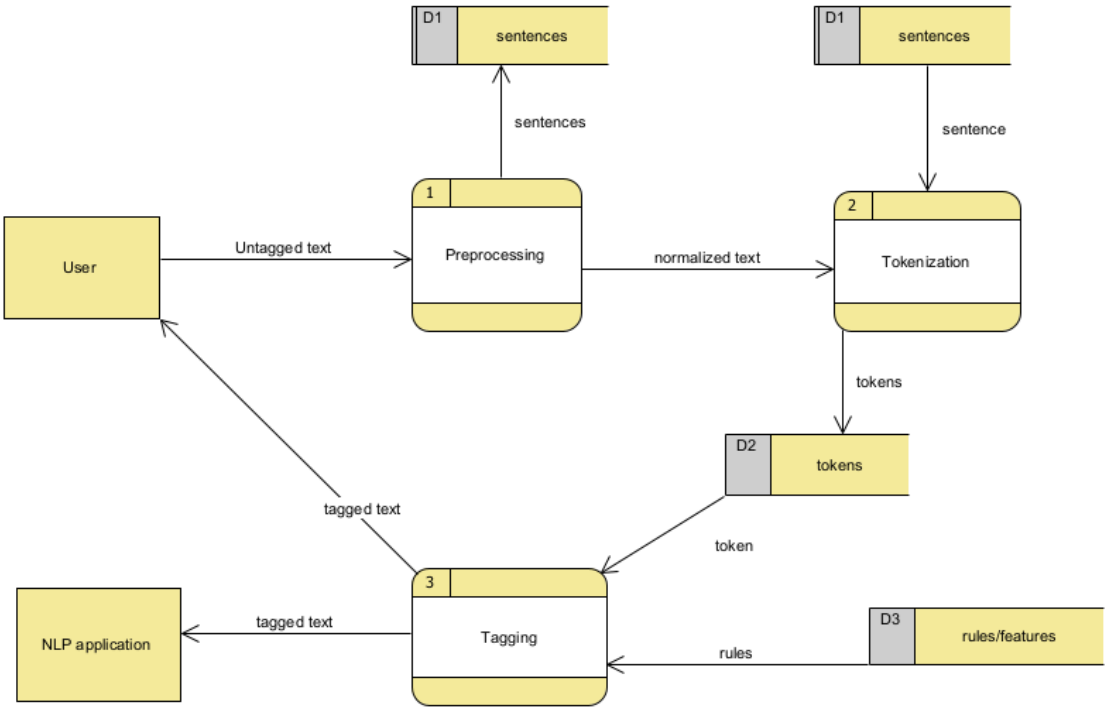


Figure 2: level 1 data flow diagram for chichewa parts of speech tagger

2.2 Description of components

This section supplies a detailed description of the constituent process of each component. Include the important data items of each process. Suitable DFD diagrams are provided to illustrate further breakdown of the process.

2.2.1 Process Analysis

The description of the processes is outlined below and it includes its pictorial DFDs

process #: 1

process name: preprocessing

attributes: sentence, token

pre-condition: the process assumes that the sentence has not been tokenized

post-condition: the sentence is cleaned ready to be tokenized

Algorithm: preprocessing

A: sentence

B: check if the sentence is cleaned

If (A = = B)

{

GOTO A

}

else {

C: remove stop words;

D: stem the sentence

E: lemmatize the sentence

}

END

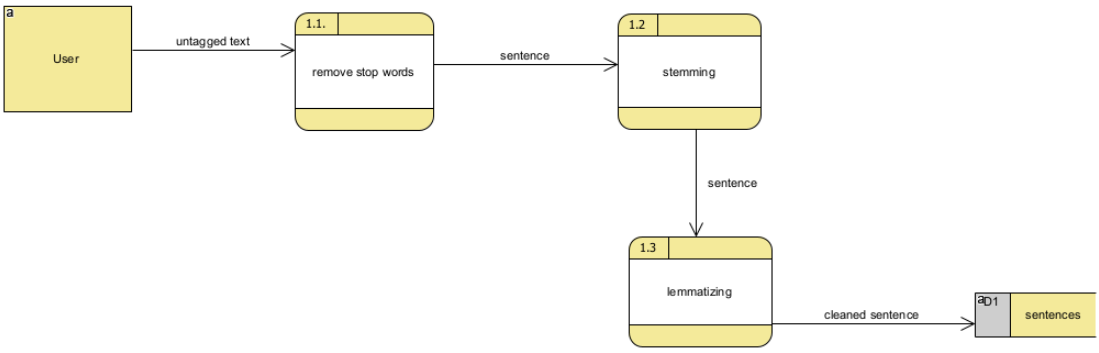


Figure 4: level 2 data flow diagram for preprocessing

Process #: 2

Process name: tokenization

Attributes: sentence, token

Pre-condition: The process assumes that the sentence is already preprocessed/cleaned

Post-condition: The sentence is tokenized ready to be tagged

Algorithm: Tokenization

A: take a sentence

B: is sentence tokenized?

If (A = = B)

{

GOTO A;

}

Else {

C: split based on white space;

D: split based punctuation;

}

END

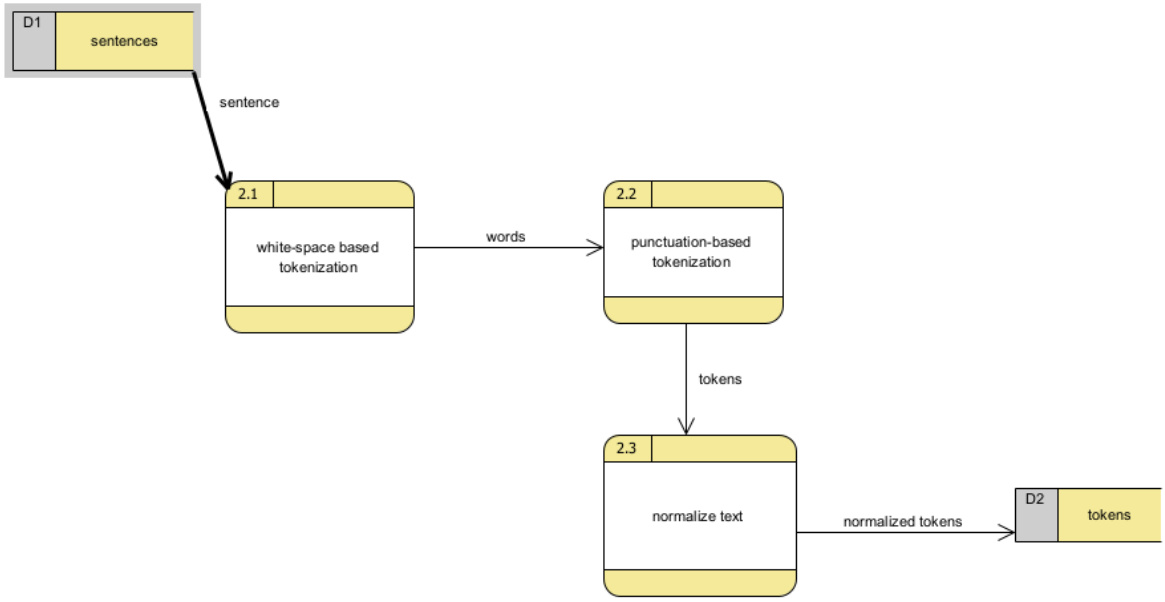


Figure 5: a level 2 diagram for tokenization process

Process #: 3

Process name: tagging

Attributes: tag, token

Pre-condition: the process assumes that the sentence to be tagged is already tokenized

Post-condition: each word of the sentence will be tagged with its corresponding part of speech

Algorithm: Tagging

A: take a token

B: check if the word is in the corpus

If (B = = True)

{

C: determine its parts of speech by

i) Applying Hidden Markov Model

ii) Applying Viterbi Algorithm

}

Else {

D: determine its part of speech by

i) Applying features/rules of Chichewa language

}

END

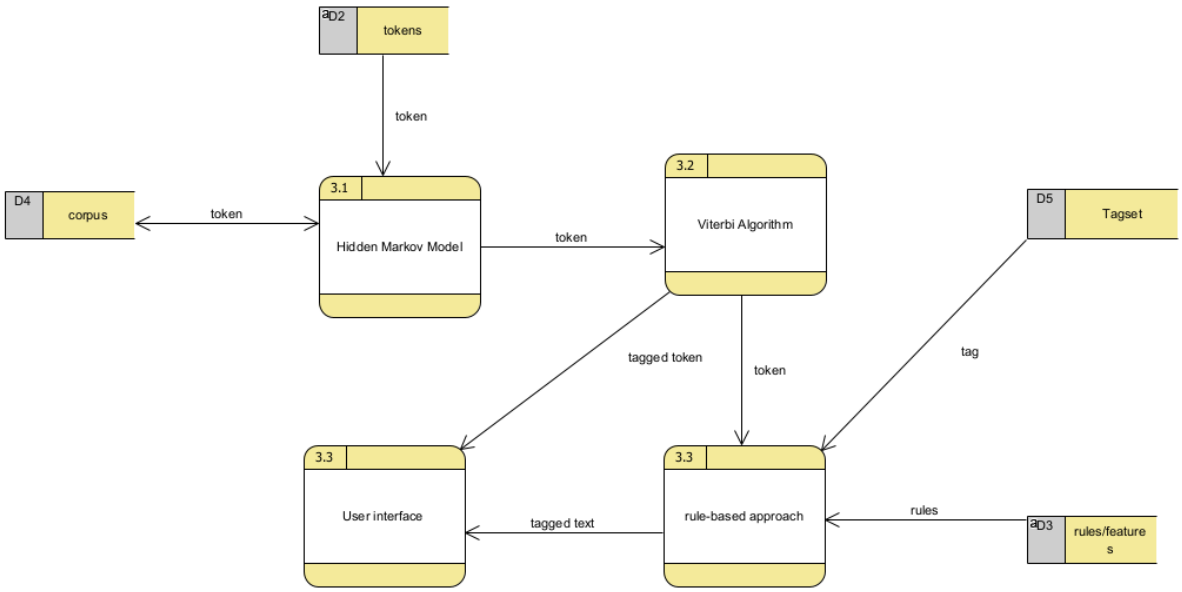


Figure 6: level 2 data flow diagram for tagging process

3.0 Data Architecture

This is a description of all persistent data items including data items

3.1 Data dictionary

It is a description of data items that are passed among components of the software in a database as shown by the table below.

column	Data type	Description	validation
userId	INT	It shows reference to user master	Numbers and characters are allowed
phoneNumber	VARCHAR	It stores phone number of a user	Only numbers allowed
email	VARCHAR	It stores email address of a particular user	Characters, symbols and numbers are allowed
name	VARCHAR	It stores the name of a particular user	Characters and numbers are allowed

password	VARCHAR	It stores the password of a user used to access the system	Combination of both characters, numbers, symbols are allowed
role	VARCHAR	It stores the type of role that the user is taking (annotator or reviewer)	Characters only are allowed
id	INT	It gives reference to a particular sentence	Only numbers are allowed
sentence	TEXT	It stores the sentence	Combination of both numbers, characters and symbols are allowed
source	VARCHAR	It stores where the sentence has come from	Only characters and numbers are allowed
progress	VARCHAR	It gives the state of the sentence whether on stemming or annotating	Only characters are allowed
tokenId	INT	It gives reference to tokens	Only numbers are allowed
index	INT	It gives the position of a particular token in a sentence	Only numbers are allowed
posId	INT	It gives reference to parts of speech tags	Only numbers are allowed

tag	VARCHAR	It stores the tag of parts of speech	Only characters are allowed
description	VARCHAR	It stores the full name of a particular parts of speech	Only characters are allowed
example	VARCHAR	It stores example of a particular parts of speech tag	Only characters are allowed

3.2 Entity Relationship Diagram (ERD)

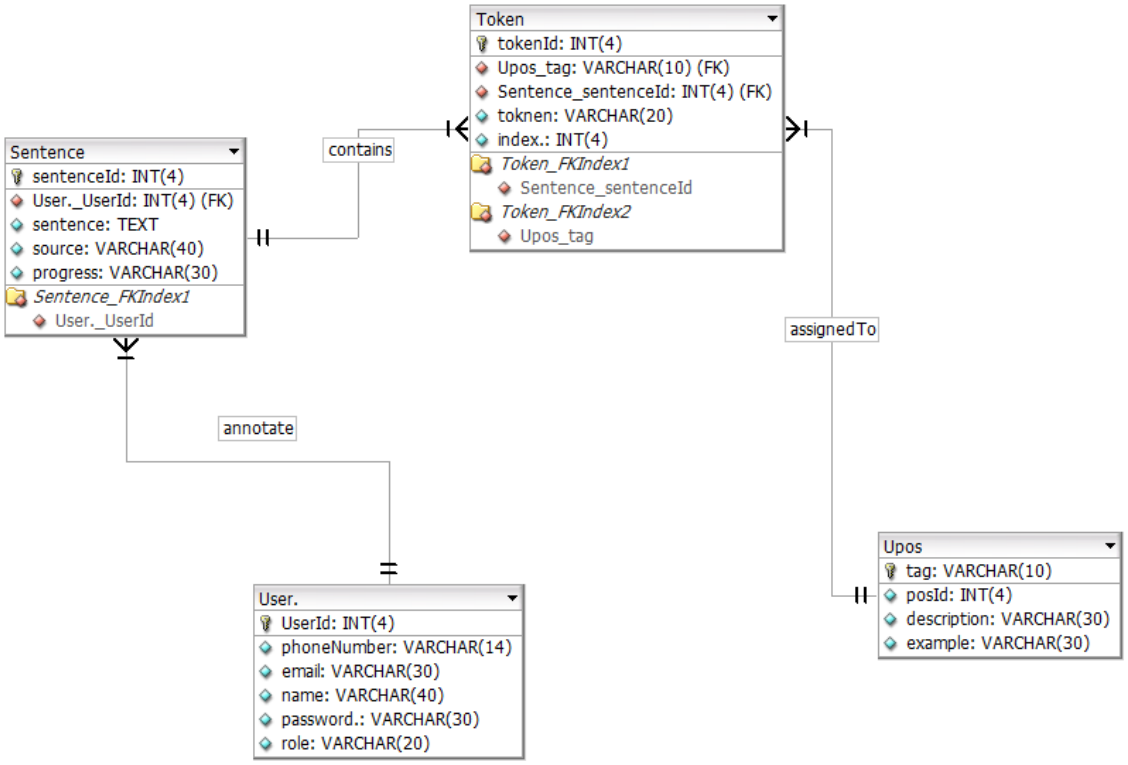


Figure 7: entity relationship diagram

3.3 Database Schema

name	type	schema
User		CREATE TABLE User (userId INT(4) NOT NULL AUTO_INCREMENT, phoneNumber VARCHAR(14) NOT NULL, email VARCHAR(30), password VARCHAR(40) NOT NULL, role VARCHAR(20) NOT NULL, PRIMARY KEY(userId));
userId	INT	userId INT(4)
phoneNumber	VARCHAR	phoneNumber VARCHAR(14)
email	VARCHAR	email VARCHAR(30)
name	VARCHAR	name VARCHAR(40)
password	VARCHAR	password VARCHAR(40)
role	VARCHAR	role VARCHAR(20)

Sentence		CREATE TABLE Sentence (sentenceId INT(4) NOT NULL AUTO_INCREMENT, userId INT(4) NOT NULL, sentence TEXT NOT NULL, source VARCHAR(40) NOT NULL, progress VARCHAR(30) NOT NULL, PRIMARY KEY(sentenceId) INDEX sentence_FKIndex1(userId));
sentenceId	INT	sentenceId INT(4)
sentence	TEXT	sentence TEXT
source	VARCHAR	source VARCHAR(40)
progress	VARCHAR	progress VARCHAR(30)
Token		CREATE TABLE Token (tokenId INT(4) NOT NULL AUTO_INCREMENT, index INT(4) NOT NULL,tag VARCHAR(10) NOT NULL, sentenceId INT(4) NOT NULL, token VARCHAR(20) NOT NULL, PRIMARY KEY(tokenId),

		INDEX Token_FKIndex1(sentenceId), INDEX Token_FKIndex2(tag));
tokenId	INT	tokenId INT(4)
index	INT	Index INT(4)
token	VARCHAR	token VARCHAR(20)
Upos		CREATE TABLE Upos (tag VARCHAR(10) NOT NULL, posId INT(4) NOT NULL AUTO_INCREMENT, description VARCHAR(30) NOT NULL, example VARCHAR(30) NOT NULL, PRIMARY KEY(tag));
posId	INT	posId INT(4)
tag	VARCHAR	tag VARCHAR(10)
description	VARCHAR	description VARCHAR(30)

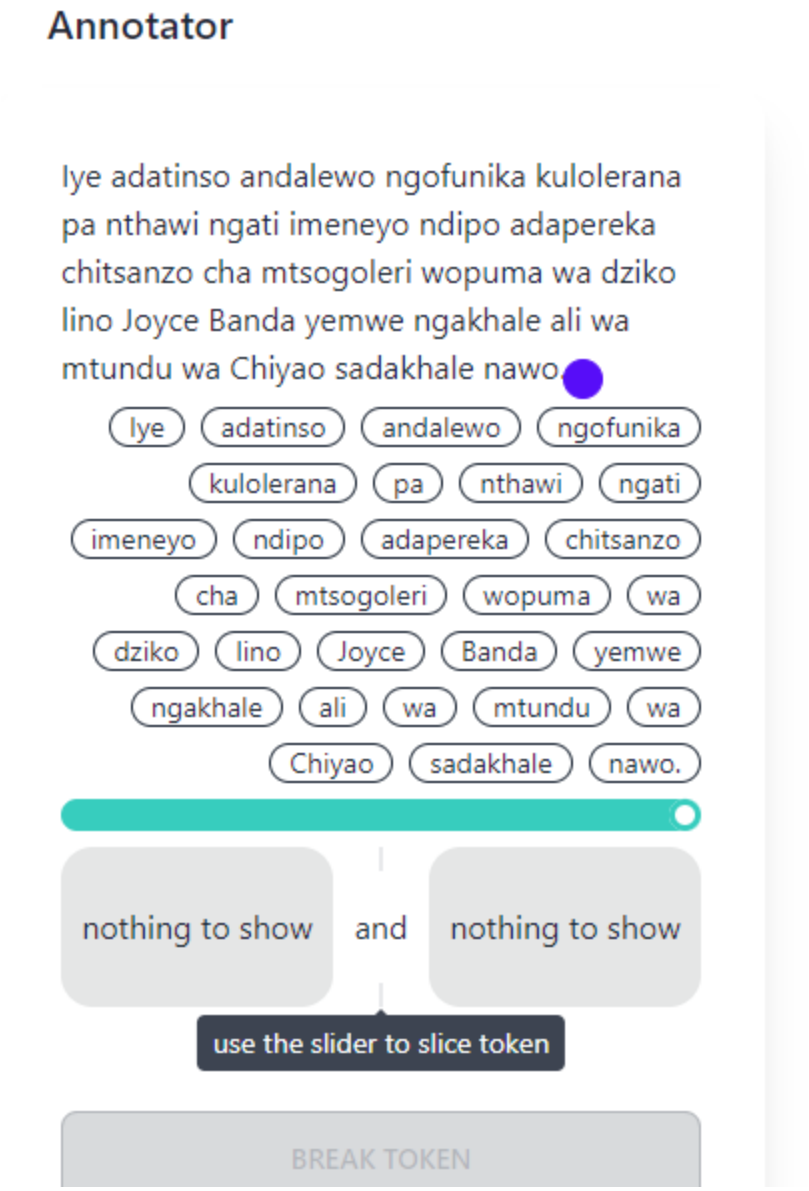
example	VARCHAR	example VARCHAR(30)

4.0 Graphical User Interface

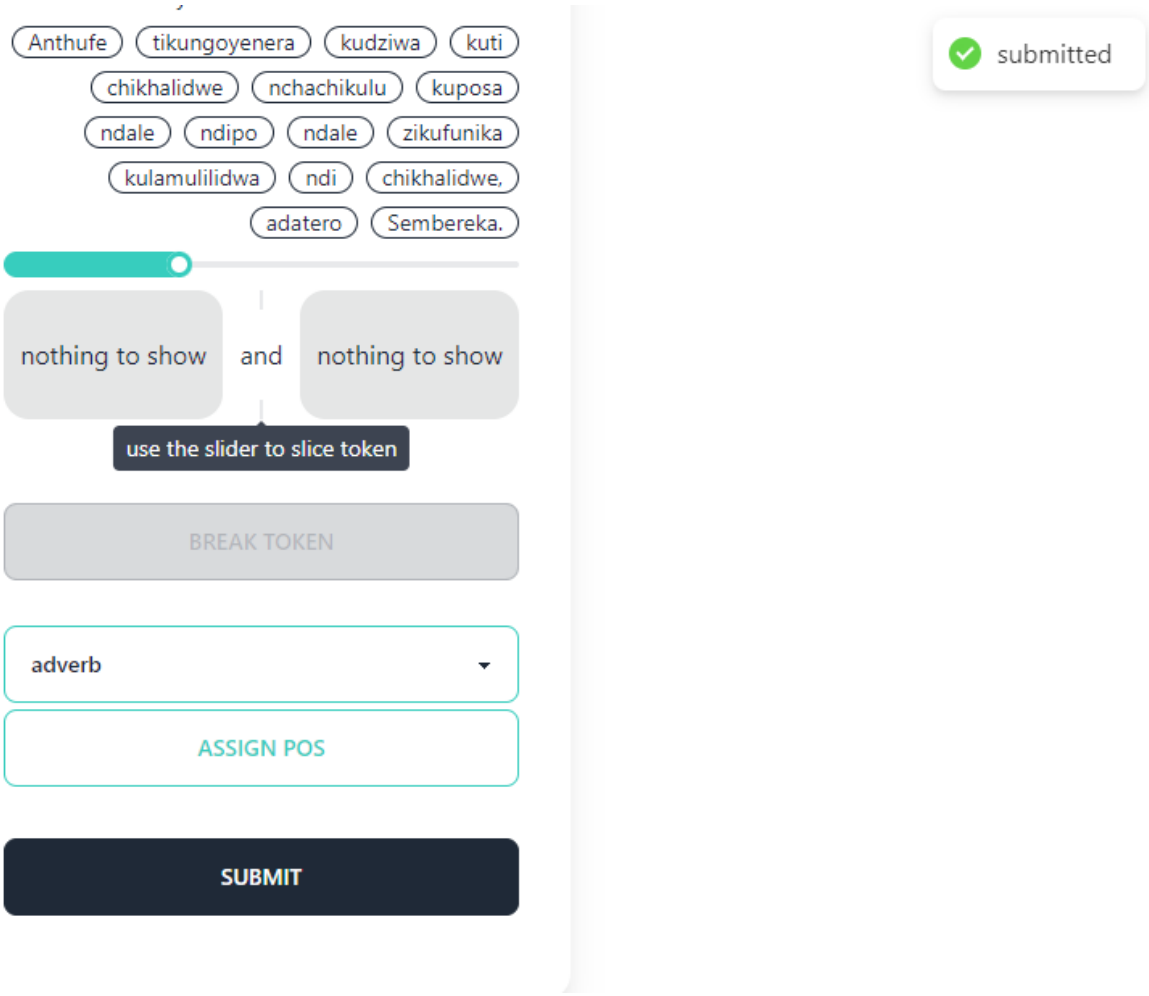
This section includes the description of the user interface of the software that is used in annotating the data of the model

Below are some of the screens shot of the software and its description

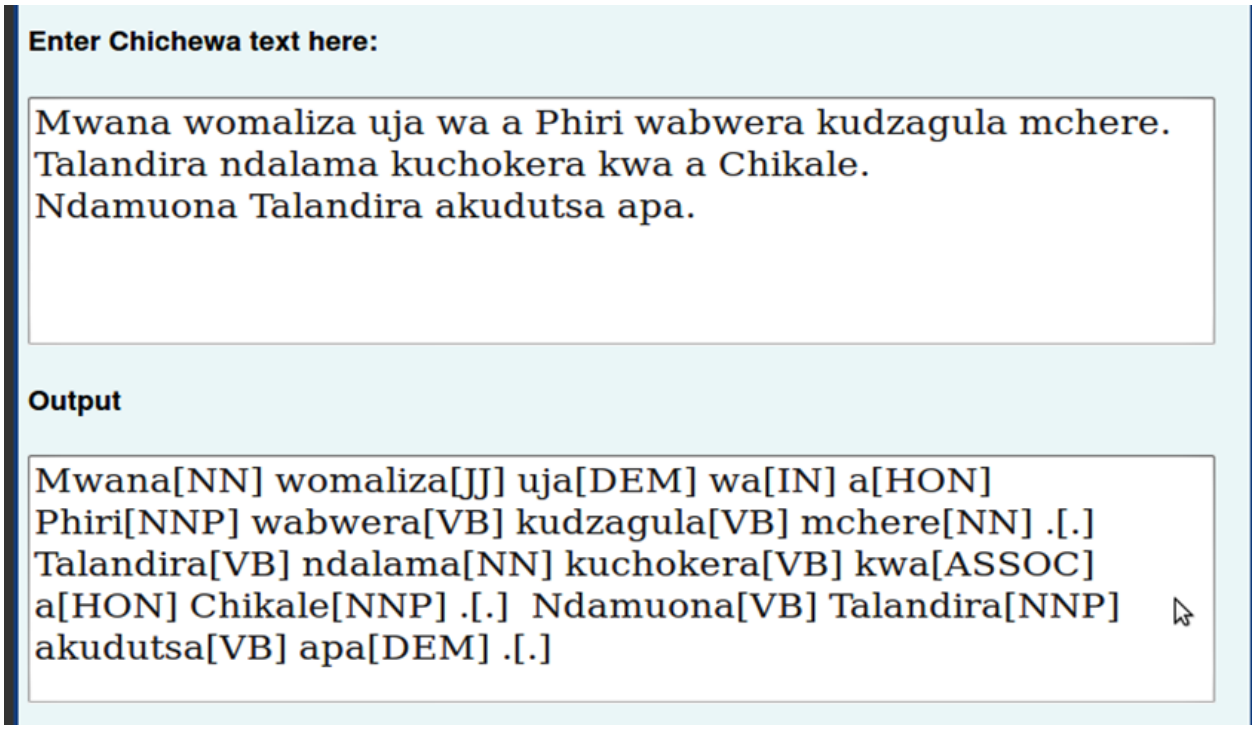
Annotator: sentence tokenization



Annotator: sentence submission after annotation



User interface of the tagger upon being implemented



5.0 Quality assurance

The system will be tested by evaluating its accuracy upon implemented. It also has web-based GUI where the results will be observed

5.1 Detailed test plans

5.1.1 Test plan for component 3

Process name: tagging

Entity name: Token

Legal input values

Token: tabwera

Illegal input values

Token: how

6.0 appendices

Algorithm: is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

POS tagger: is a piece of software that reads text in some language and assigns parts of speech to each word (and other token)

Rule based approach: is one of tagging approaches that uses rules/features to tag words in a text

Stochastic/probabilistic approach: is the tagging approach that words in a text based on the probability of a word belongs to a particular tag or based on a sequence of preceding / succeeding word

Corpus: a collection of written texts especially the entire works of a particular author or a body of writing on a particular subject

Database schema: is the blueprint of database which describes how the data may relate to other tables or other data models

Annotator: is a note or comment added to a text to provide explanation or criticism about a particular part of it.

7.0 GLOSSARY

Term	**Definition**
HMM	Hidden Markov Model
NLP	Natural Language Processing
POS	Parts Of Speech
DFD	Data Flow Diagram
GUI	Graphical User Interface

8.0 References

[1] J. M. D, "computation Linguistic and speech Recognition," pp. 20-100, July 2002.

[2] I. Mashhad, a study on the part of speech, Khavan Higher-Education Institute, 2011.

[3] A. voutilainen, part of speech tagging, New york, 2012.

[4] Z. A. A. Altabba M, "An arabic Morphological Analyser and Part of Speech tagger," Mumbai, 2010.