Suffix-stripping Algorithms and Transducers for the Fulani Language

Zouleiha Alhadji Ibrahima *zouleihaalhadji@gmail.com*
Department of Mathematics and Computer Science, Faculty of Science, the University of Ngaoundéré

Dayang Paul *pdayang@univ-ndere.cm*
Department of Mathematics and Computer Science, Faculty of Science, the University of Ngaoundéré

Kolyang *Kolyang@univ-maroua.cm*
Department of Computer Science, Higher Teachers Training College, the University of Maroua;

Guidana Gazawa Frederic *fguidana@gmail.com*
Department of Mathematics and Computer Science, Faculty of Science, the University of Ngaoundéré

Abstract

Because of the large and constantly increasing amount of information available on the Internet, users are facing diverse challenges and difficulties while trying to satisfy their needs. In fact, the objective of today's information retrieval systems is no longer accessing information but the search and filtering of relevant information. The language used for searching information plays a major role. If we consider resource scarce local or national languages, the situation becomes even more challenging. Many African languages fall into the group of resource scarce languages. Therefore, there is a need to explore and build more specialised information systems that enable speakers of African languages to discover valuable information across linguistic and cultural barriers. As one of the most dispersed languages in Africa, the Peul also called Fulani language suffers from a significant handicap in its computerisation and automatic processing due to the inexistence of digital and linguistic resources. Considering the fact that a devoted care and attention to conserve, guarantee the sustainability of languages is important, few studies and computerisation works have been carried out on African Languages such as Fulani. The aim of this work is to lay some bricks towards tools for the automatic processing of the Fulani language. This language belongs to several dialectal areas and there are almost no digital documents of the Fulani language of the Adamaoua dialectal area. The originality of this work is among others the digital processing of Noye Dominique Fulani dictionary from North Cameroon; we then studied stemming approaches for Fulani words using transducers that clearly show how to remove classifiers from words in order to obtain the stem. To do so, we have grouped all the classifiers that are suffixes in number: singular and plural and by degree of classifiers. An example of the process of removing a suffix has been described in this article. Up to date, no research work has been done aiming at processing the Fulani language or native African languages similar to Fulani. In fact, the stemming approach is crucial in all information retrieval systems because it allows the translation and the classification of documents as well as indexing of words. To specify the stemming approaches, we have adapted the stemming algorithms of Lovins and Porter to the Peul language, knowing that they are the best known in literature and they have the advantage of being applied to other languages. Finally, the evaluation of these stemming methods was done using the method of Christ Paice. Based on the principle that words sharing the same stem are likely to share a unity of meaning, we undertook a morphological analysis of 5186 Fulani words from the Fulani dictionary of Dominique Noye. The results obtained from this method by

calculating the error rates of over-stemming, under-stemming and truncation errors have shown that both algorithms are efficient for the stemming of Fulani language.

Keywords: Peul, Fulani, Suffix-stripping, Stemming, Linguistic, Transducers.

---

1. INTRODUCTION

Human societies are multilingual, and globalization calls for the increased use and acquisition of several languages for communication, mediated or not by information technology (Le 2019).

For several years now, electronic documents have been increasing both on the Internet and in corporate intranets. The amount of information generated in different languages and distributed via the World Wide Web (WWW) and social networking platforms is growing exponentially. As the Internet has become ubiquitous and widespread, its users have become linguistically more diverse and culturally more heterogeneous. However, many users are being discouraged by the slowness and inefficiency of the available traditional search engines (Omri 2004). One of the common causes of these failures is the almost complete lack of linguistic analysis that considers a wide range of languages in query and document processing (Paternostre et al. 2012). Current information retrieval systems focus mainly on European languages, such as English, which remains the most widely used language on the web. Harrathi et al. have shown that "currently the proportion of Internet users using a language other than the European language is 44.5%" (Harrathi et al. 2009). Automatic language processing tools help in the writing, translation and manipulation of documents. These tools are part of information retrieval systems. Only 1000 of the world's 7000 native languages are supported online (Cefan 2021). This is due to the fact that most of these languages are not yet digitized. In African and Asian countries, access to online resources and services is particularly limited by barriers such as languages, coupled with a lack of appropriate infrastructure. It is therefore more urgent than ever to explore and build more specialized information systems that allow speakers of African languages to discover valuable information across language and cultural barriers. For this reason, when one wishes to initiate or improve the computer processing of a poorly endowed language, it is logical to create the basic resources before tackling the tools. These resources are usually lexicons and/or corpora, annotated or not, monolingual or parallel (Kevers et al. 2019) However few research topics address the issue of processing African languages, which have limited resources. This is the case of the Fulani language, which has a total population of over 40 million people in more than 15 countries (Cefan 2021). In this paper, we present methods for the stemming process in the Fulani language of the Adamaoua dialect area. These methods are based on Porter algorithm and Lovins algorithm. Then, we defined transducers to show how to remove suffixes from words. To reach this purpose, we have previously processed a morphological analysis of the words in Fulani and we have digitized the Fulani dictionary of Dominique Noye from North Cameroon.

The final objective is to develop tools for automatic processing of documents in the Fulani language for effective use in information retrieval systems. The method used here is the application of rules that will allow the removal of classifiers from words in order to group semantically close words according to their similarities and to use them for the development of applications such as information retrieval and classification systems. In other words, the aim is to give a digital identity to the Fulani language. Thus, throughout our study, we will present the state of the art on the automatic processing of natural languages and stemming algorithms, we will deal with the morphological description of the Peul language. Then, we will present how to use a transducer to remove suffixes from words. Finally, we will present the Porter and Lovins (Porter 1997; Lovins 1968) algorithms adapted to the Peul language and the results obtained after evaluation by the Christ Paice method (Paice 1994).

2. LITERATURE REVIEW

In linguistics, stemming is a process of transforming or reducing inflicted words to their word stem.

Stemming deals with families of derivationally related words with similar meanings represented using single term. A stemming algorithm is a procedure that reduces all words with the same stem to a common form by stripping its derivational and inflectional suffixes (Tesfaye et al. 2010).The techniques used to do this are generally based on a list of affixes (suffixes, prefixes) of the language under consideration and on a set of a priori constructed stemming /suffix-stripping rules that allow a given word, to find its stem. Stemming is a widespread method of word normalization, designed to allow the matching of morphologically related terms such as "clusters" and "clustering". For example, in English language, a typical word contains a stem that refers to a central idea or meaning, and some affixes are usually added to modify the meaning and/or adapt the word to its syntactic role (Porter 1997; Lovins 1968). Search engines use stemmers to improve information retrieval (Omri 2001). Using Stemming, many contemporary search engines associate words with prefixes and suffixes to their word stem, to make the search broader. The process aims to ensure that the greatest number of relevant matches is included in search results (Tesfaye et al. 2010). Keywords in a query or a document are represented by their stems rather than the original words. Multiple variants of a term can thus be grouped in a single representative form, which reduces the size of the dictionary, i.e. the number of distinct terms needed to represent a set of documents. A small dictionary saves both space and execution time (Boukhari 2013). Two main families of stemmers are present in the literature, algorithmic stemmers and those using a dictionary.

On the one hand, algorithmic stemmers are often faster and extract stems from unknown words using some defined rules. However, they will have a higher error rate, grouping together a set of words that should not be (over stemming errors or under stemming errors). On the other hand, the dictionary-based approach does not produce errors on known words but on those, which do not exist in the list. It is slower, and still requires the removal of suffixes before searching for the corresponding stem in the dictionary. Each algorithm has its own steps and different rules. In addition, we can categorize the stemming algorithms into three main groups: truncating methods, statistical methods and mixed methods. Truncating methods serve to remove affixes from a word according to the rules of the language being processed. In this family of stemmers, we have four main stemming algorithms: Lovins, Porter, Paice/Husk and Dawson Stemmer. Statistical methods are stemmers which are based on statistical analysis and techniques. These methods remove affixes after implementing a statistical procedure (Jivani 2011). In this type of stemmer, one is called to determine a distance. The measure of distance between words, is a way of representing numerically the difference between two words, or more generally two strings of characters. Therefore, a low distance value is assigned to a pair of character strings, which are morphologically similar, and a high distance value to words without morphological relationship (Majumder et al. 2006). We have three algorithms: N-gram stemmer, HMM (Hidden Markov Model) stemmer and YASS (Yet Another Suffix Stripper) stemmer. Mixed method is characterized by three different concepts: inflection and derivation methods, corpus-based stemmers and context sensitive stemmers. The corpus must be very large to develop these types of stemmers (Jivani 2011).

On the other hand, stemming approaches have been developed for the Arabic language: manual dictionary construction, light stemming and morphological analysis (Younoussi et al. 2007). For manual dictionary construction, Al-Kharashi & Evens worked with small text col-lections for which they manually built dictionaries of stems and pseudo-stems (stems) for each word (Al-Kharashi et al. 1994) this approach is not feasible for a large corpus but also the addition of a new word stems difficult to manage. Light stemming on the other hand is a process of removing suffixes and prefixes without worrying about infixes to recognize the stem. Algorithms realized for this approach have the same flaws because they are based on a predefined list of prefixes and suffixes that are supposed to be removed when they are attached to the word whose root is being searched (Allah et al. 2010). Based on a list of affixes, many erroneous or sometimes ambiguous results are generated, either because the letters considered as affixes are removed even though they are original, or because some affixes are not removed because the algorithm (Younoussi et al. 2007) does not consider them. Morphological analysis consists in grouping together forms

sharing the same stem.It is done in two steps: the first one consists in applying light stemming and the second one consists in recoding which adds predefined endings to the obtained stems in order to have a good stem. Morphological Analysis forms an essential pre-processing step in natural language processing (Mahyoob 2018).The drawback of this method, like the light stemming, is that it is based on the suppression of predetermined affixes before applying the notion of scheme. In (Tesfaye et al. 2010), a stemming system for Afan Oromo is presented. This system takes as input a word and removes its affixes according to a rule-based algorithm.

In order to reduce the rate of error and ambiguity, generally due to the removal of affixes, a stemming technique for the Arabic language is proposed, the approach is based on finite state automata (Younoussi et al. 2007). Arabic nouns and verbs are derived from stems by applying patterns, the latter often involving the addition of infixes, deletion or replacement of the letters of the stem. This method of stemming has two steps: a pre-processing step preceding the stemming, which consists in making a series of modifications in order to make the text conform to certain standards. For example, removing punctuation, removing empty words, replacing some letters with others at the beginning or end of words. Then comes the processing phase, at this level, the author enters the word into the five automata he has de-signed corresponding to all the schemes of the Arabic language. In each automaton, if the word manages to find a path to reach the final state, he retrieves the root and checks its validity in the list of valid stems created by Kareem Darwish, which contains about 10,000 stems (Darwish 2002). If the found root is not valid or if the final state cannot be reached, the first letter is considered as prefix and it starts again, once the first letter is removed. The difficulty with this method is that in most cases the approach returns several stems, while only one is sufficient, and the rest is practically out of context. In (Samuel et al. 2018), a rule based stemming algorithm that conflates Kambaata word variants has been designed for the first time. The algorithm is a single pass, context-sensitive, and longest-matching designed by adapting rule-based stemming approach.

However, it is important to note that an automatic language processing technique is specific to each language and not to a group of languages. Since the languages rely on different grammar, spelling, conjugation etc. That is why for the stemming of the Fulani language for which no method exists until today, the above methods have not been applied correctly but these methods gave us the idea of how to do a stemming. Fulani is a language with a complex morphology, very different from the structure of other languages such as Arabic and Amazigh. It has no prefixes but rather a well-defined list of suffixes. In order to obtain the stem of Fulani words, we found, after a morphological analysis of the words, that it is necessary not only to remove suffixes at the end of the words, but also to remove double vowels and short vowels. Among the above-mentioned stemming methods, we have chosen to adapt the Porter and Lovins (Porter 1997; Lovins 1968) algorithms to the language because they are efficient compared to other methods and they are adaptable to other languages.

3. THE FULANI LANGUAGE

The Fulani language, commonly called Fulfulde, is one of the African languages belonging to the family of Niger-Congo (Heine et al. 2000) which counts more than 1500 languages, the largest in the world. It is a language with a nominal class and/or stem (Mohamadou1991); the different names of the language are: Pulaar (FuutaTooro), Pular (FuutaJaalo), Fula (Ghana), Fulfulde (Adamaawa and Maacina). The man who speaks this language is called Peul (French), Fallata (Arabic), Fulah (English), Fulaani (Hawsa); but the Fulani person is called pullo/fulbe. Fulfulde is divided into six dialectal areas: Foulaadou (Southern Senegal), sokkoto (Niger and Central Nigeria), FuutaJaalo (in Guinea Conakry, Guinea Bissao, Sierra Leone and Liberia), fuutatooro (Senegal, Mauritania and western Mali), Maacina (central Mali, Ghana and Burkina Faso) and finally Adamaawa (Southern Nigeria, Southern Chad, Cameroon, part of Northern Central Africa and Sudan) (Cefan 2021; Noye 1974).

3.1 The Alphabet and the Classifiers

The Fulani alphabet FULCOM (Fulani commission) is part of the International Phonetic Alphabet (IPA). It was adopted during a conference on the orthographies of African languages held in

Bamako, Mali in 1966 by United Nations Educational, Scientific and Cultural Organization (UNESCO) (Tradlibre 2021). The Fulani language has 27 consonants (b, ɓ, c, d, ɗ, f, g, h, j, k, l, m, n, ny, ŋ, p, r, s, t, w, y, ƴ,' mb, nd, ng, nj) and 5 vowels (a, e, i, o, u). But, there are also long vowels: aa, ee, ii, oo, uu in the Fulani language. It can be noted that compared to French for example, the Fulani language does not have the following letters: q, v, x, z (Amidou 2009; Diallo 2015). The Fulani language has several points in common with the Bantu languages, namely the nominal classes. These classes are marked by suffixes and not prefixes. Another peculiarity of Fulani is the substitution of the initial consonant when the noun changes from singular to plural.

Example: *O raarii→ɓendaarii*
Fowru→pobbi

The Fulani language ignores the masculine/feminine opposition but it distributes nouns into a large number of classes. On the one hand, it distinguishes between personal names which have one class for the singular (o) and another for the plural (ɓe), and on the other hand, names of animals or things which are divided in the oriental dialect into eighteen classes for the singular and five classes for the plural. That is a total of twenty-five classes (Noye 1990). However, each nominal class can have 1st, 2nd, 3rd or 4th degree classifiers.

3.2 Verbal forms
The verb is a variable word, which can take different forms depending on the desired meaning. Like nouns, pronouns and adjectives, it varies according to number (singular or plural) and sometimes gender (masculine or feminine). But it also varies according to the person, the time, the mode and the voice. In the Fulani language, verbs take three voice forms (Mohamadou 2014; Conjugaison 2021):

- Verbs of the active voice: These are verbs that end with "*go*" or "*ugo*", for example: *yahgo, winndugo*

- Middle-voiced verbs: These are verbs that end in "*aago*", for example :*yiiwaago*

- Passive voice verbs: These are verbs that end in "*eego*", for example :*mooreego*

The different voices do not influence the stemming algorithm because it removes suffixes and double vowels that come before the suffixes. A verb conjugated in the active, passive or middle voice will have the same stem once the stemming process is completed.

For example, the following words *yiiwgo*, *yiiwaago* and *yiiweego* will have as stem "*yiiw*".

3.3. Derivatives
A derived word is formed by adding one or more affixes (prefixes or suffixes, welded) to a lexical morpheme know as base; the ultimate, minimal base is called radical. Derivation is an action that consists in forming from the stems different verbs that will indicate certain modalities of action. Indeed, while French uses prefixes such as "ré-" and "pré-", or "re-" and "pre-" for English, Fulfulde uses the derivation suffix which is filed after the stem in order to form a derived verb. This derived verb has the same ending as simple verbs. With the help of these derivation suffixes, different nouns, places, instruments, agents, etc. can be given or formed. The Fulani language of the Adamaawa dialectal area has fifteen (15) derivation suffixes (Arnott 1960; Taylor 1953) which are:

-(i)n- : causative or factive derivative, for example : *janngingo*, *ƴummingo*.

-an- : beneficial derivation, for example: *defango*.

-(i)r- : instrumental derivative or derivative of manner, for example: *mi winndiriijaawal / mi winndiriibataakewolbinndirgo*l.

-or- : instrumental derivative or derivative of manner, for example : *mi jooɗorakejaawal / mi jooɗorakejooɗorgal*.

-(i)d- :associative or fine derivative, for example : *janngidgo*.

-od- : associative or fine derivative, for example : *waalodaago*.

-(i)ɗ- : verbalizer derivative, for example : *goongɗugo*, *semmbiɗgo*.

-w- : verbalizer derivative, for example: *ranwugo*.

-oy- : distant derivative, for example: *ñaamoygo*.

-(law)- : accelerator derivative, for example: *yarlawgo*, *winndilawgo*.

-(i)kin- / -kikin- : simulative derivative, for example : *ñaamkikingo*, *dillikingo*.

-(i)ndir- : reciprocity derivative, for example: *hoofnindirgo*, *fiyindirgo*.

-ootir- : reciprocity derivative, for example : *yiɗootirgo*, *enɗootirgo*.

-it- : repetitive, inversive, simulative derivative, for example : *maɓɓitgo*, *jaɓɓitaago*, *ɗaɓɓitgo*.

In fact, Fulani stems associated with derivatives are called complex stems.
We have noticed that each time we derive a verb, the stem of the initial verb will be different from the stem of the derived verb.

4. TRANSDUCERS RELATED TO STEMMING ALGORITHMS

A finite transducer (also called a finite state transducer) is a finite automaton with outputs. It is an extension of finite state automata. They operate on words using an input alphabet. Instead of accepting or rejecting the word, they transform it, sometimes none deterministically, into one or more words of an output alphabet. This allows transformations of languages, and various uses such as syntactic analysis of programming languages, and morphological analysis in linguistics. Sextuplet (Francois 2007) defines it;

- T=(**Σ**1,2,Q,q0,F,δ) where
- **Σ**1, **Σ**2 is a finite set of symbols (input and output alphabets respectively);
- Q is a finite set of states;
- q0 ∈ Q is the initial state;
- F is the set of final statements;
- $\delta$is a partial function of (Q X$\sum_1^*$ X$\sum_2^*$) in 2^Q.

The machine starts in the start state q0, the machine will transit from state to state with the data according to the transition function δ. Finally, the machine accepts data if the last input of thisdata causes the machine to halt in one of the accepting states (Mahyoob 2018).

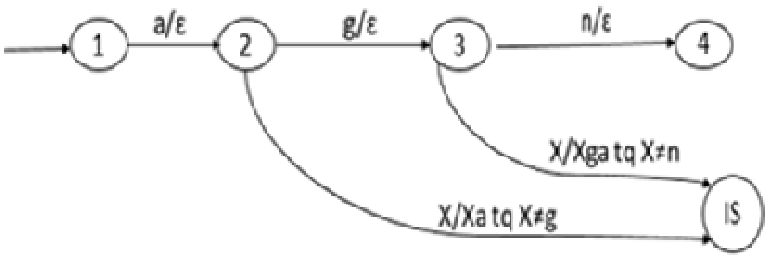
Our work is based on the corpus extracted from Noye Dominique is Foulfoudé-Français dictionary. In this work, we grouped the classifiers in number (singular and plural) and by degree of classifiers. Thus, we counted four groups of classifiers in the singular from the 1st degree to the 4th degree and 3 groups of classifiers in the plural from the 2nd degree to the 4th degree. As shown in table, we found that words in the plural do not have a classifier of the 1st degree.

	Classifiers 1st degree	Classifiers 2nd degree	Classifiers 3rd degree	Classifiers 4th degree
Singular	nde, ndu, nge, ngo, ngal, ngel, ngol, ngum, ndi, nga, ngu	de, du, ge, go, ko, gal, gel, gol, gum, ɗum, kal, ki, ga, gu, ɗam, ka	jo, ko, wo, ɗo, re, ru, ye, wo, wal, wel, yel, wol, wum, yum, rum, jum, hal, hi, ri, wa, wu, jam, ha	e, u, o, al, el, ol, um, i, a, am.
Plural		ɓe, kon, koy, ko, ɗe, le, ɗi, li	en, hon, hoy,ho, je, ji.	on, oy, o, e, i

TABLE 1: Grouping of Classifiers by Number and Degree.

Transducers were used in this work. Each level of classifiers includes several classifiers.

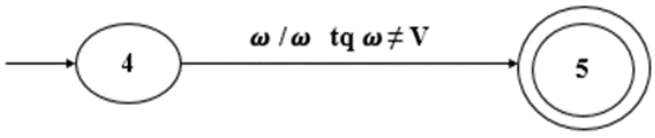
To explain how these transducers work, let us take the example of a word ending in "*nga*" (Figure 1). When the word enters the algorithm, the reading goes from right to left. If the transducer reads 'a', it deletes it and the same if it reads 'g', it deletes it. On the other hand, if after reading 'a' it reads another letter of the Fulani alphabet other than 'g' it writes that letter followed by 'a' and falls into the initial state of the next classifier level. Otherwise it continues reading 'n', and deletes it, if it reads another letter other than 'n' then it falls into the initial state of the next group of classifiers. This is how the transducer goes through the letters to remove the classifier associated with the word. After removing the "*nga*" classifier, when the algorithm reads a letter of the Fulani alphabet, it writes this letter (Figure 2) and at the end of the reading process we obtain the stem of the word.



IS: Initial state

FIGURE 1: Transducer « nga ».

The final transducer leaving state 4 to acceptance state 5 is as follows. ω ε Σ (Σ being the set of Peul alphabets).



ω ε Σ (Σ étant l'ensemble des alphabets peul)
V (ensemble des voyelles)

FIGURE 2: Final Transducer.

5. APPLYING ALGORITHMS TO THE FULANI LANGUAGE

The Peul language has a particularity on the shape of the words; however, it's stemming will be more adapted to the algorithms conceived on the basis of rules. Among these algorithms, Lovins and Porter are the most efficient being used and quoted in literature. Thus, we wanted to apply them to the Fulani language to compare their efficiency.

5.1 Application of the Lovins Algorithm to the Peul Language

The Lovins "stemmer" in English language has 294 endings, 29 conditions and 35 transformation rules and where each ending is associated with one of the conditions. In the first stage, if the longest ending found satisfies its associated condition, that ending will be discarded. In the second step, the 35 rules are applied to transform the ending. The second step is performed if an ending is deleted in the first step or not.

In the case of the Fulani language, we will modify the two steps of Lovins to obtain the stem of the words.

5.1.1 Description of the Following Diagram

The flowchart below shows the different phases that summarize the whole sequence of stemming of a Fulani word with the Lovins algorithm.

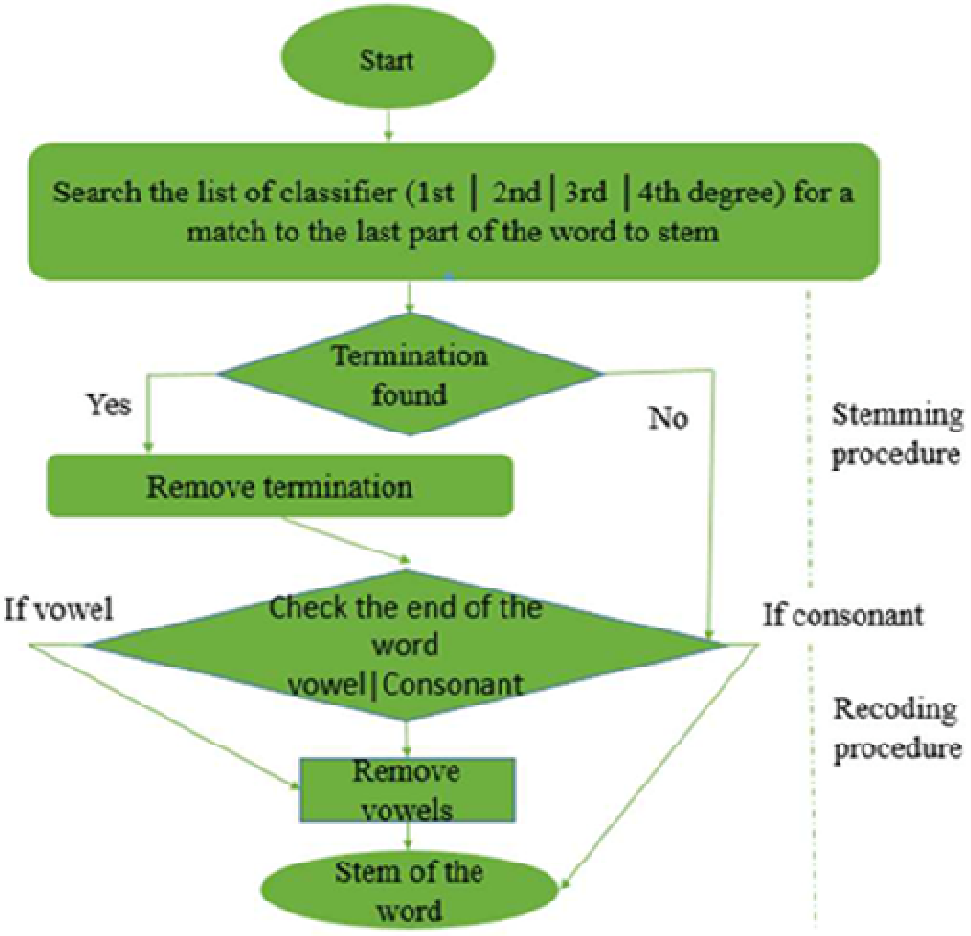


FIGURE 3: Steps for adapting the Lovins algorithm to the Fulani language.

Lovins stemming procedure is divided into two steps: a stemming procedure and a recoding procedure. We have adapted it to the Fulani language according to the diagram above.

Stemming procedure
Step 1: enter the word in the algorithm
Step 2: search the list of classifiers for a match to the last part of the word to stem
Step 3: if the termination is found, delete the classifier otherwise go to the next step
Recoding procedure

Step 4: check the end of the word vowel or consonant
Step 5: If vowel, remove vowels. If consonant, return stem of the word

The pseudo code of the Lovins algorithm applied to the Fulani language has the following structure.

// Loop while we have word in file

```
Word ← file.readline
has_classifier← true
Whilehas_classifierdo:
//get word classifier
                Classifier ← word.Classifier()
                //switch into each case
switch (classifier):
case is 1st degree classifier then
RemoveListFirstDegreeClassifier()
case is 2nd degree classifier then
RemoveListSecondDegreeClassifier()
case is 3rd degree classifier then
RemoveListThirdDegreeClassifier()
case is 4th degree classifier then
RemoveListFourthDegreeClassifier()
                    default
has_classifier←false
if (word.vowel() )
Word.RemoveVowel()
Else (write stem to new file)
endIf
endswitch
      endwhile
```

After applying Lovin's algorithm on 5186 Fulani words, we obtained 3180 distinct stems, that is a reduction of the number of words to 61.32%.

5.2. Application of Porter Algorithm to the Fulani Language

Considering the case of the English language, the algorithm developed by Porter is composed of about fifty rules of stem/desuffixation classified in seven successive phases (treatment of plurals and verbs in the third person singular, treatment of past tense and progressive, ...). The words to be analysed go through all the stages and, in the case where several rules could be applied to them, the one with the longest suffix is always chosen. The stemming or desuffixation is accompanied, in the same step, by some recoding rules (Boukhari 2013). It is based on the notion that suffixes in the English language (about 1200) are mainly composed of a combination of smaller and simpler suffixes (Jivani 2011). This algorithm gives the best output as compared to other stemmers, less error rate. In addition, the Snowball framework designed by Porter is a language independent approach to stemming. For the Fulani language, we will try to customize and adopt the Porter approach to consider some its particularities as described above for the English language.

5.2.1 Word Measure

Since the porter algorithm has the advantage of being applied to other languages, we have adapted it to the Fulani language by applying some modifications and we have added a new transducer-based method to remove the classifiers and get the root of the words. Any name in Fulani has the following form: Stem + nominal-class-classifier.

Fulani words are in the form:

- CVC...C,example: Mabbol
- CVC...V, example: Nasiya
- VCV...C,example: agodangel
- VCV...V,example: eggititgo
-

This study was made after studying the morphology of Fulani words in Noye Dominique is Foulfoudé-French dictionary (Noye, 1990).

The VC couples then correspond to the pseudo-syllables of the word. The number of pseudo-syllables of the word is called the 'measure of the word' and is noted « m ».

For m=1, we can have: *sondu* with V=oand C=nd; *saaŋge* with V=aa and C=ŋg.
For m=2, we can have: *peetum* with V=ee, C=t and then V=u and C=m.
For m=3, we can have: *aduwoyru* with V=a, d=m ; V=u, C=w ; V=o, C=yr.
For m=4, we can have: *nyaameteeɗum* with V=aa, C=m ; V=e, C=t ; V=ee, C=ɗ ; V=u, C=m.
Etc.

5.2.2 Transformation Rules
As in Porter, our algorithm proceeds in several steps through which the words to be processed pass successively. The transformation rule is as follows:

(Condition) S1 → S2

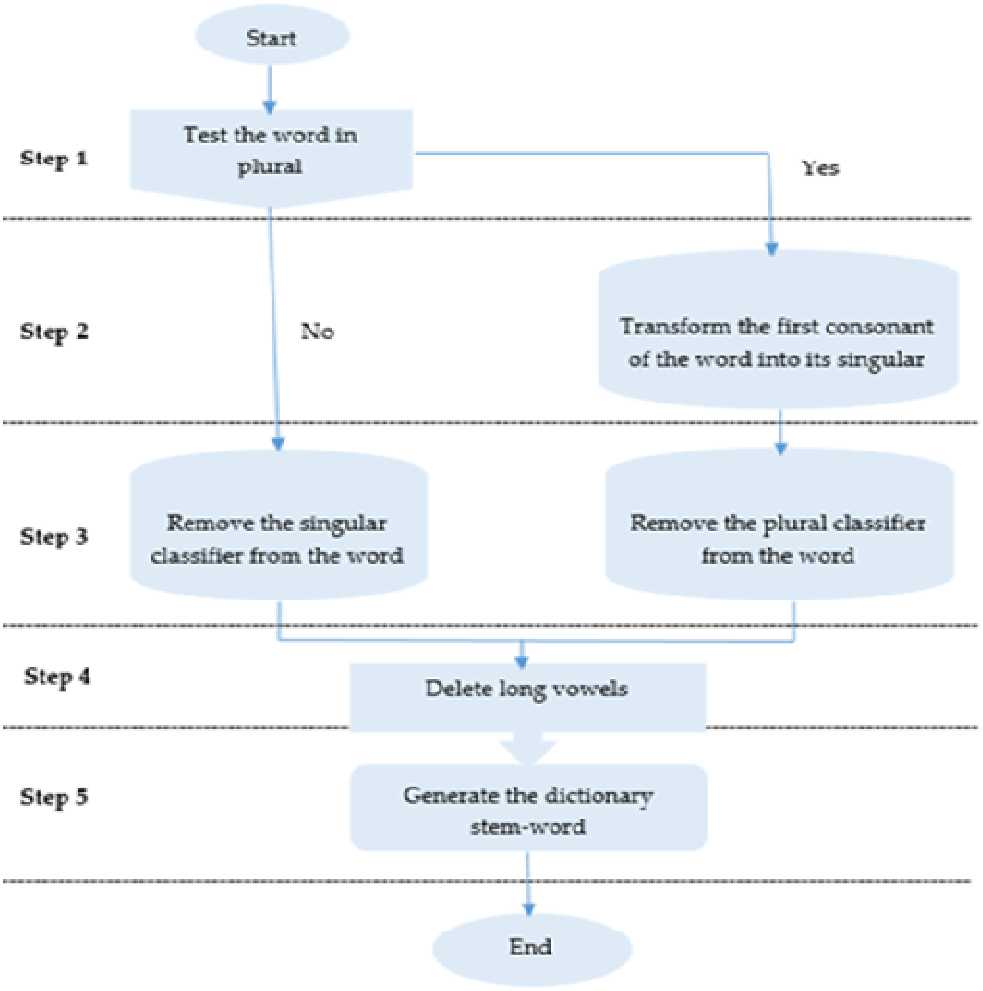
Starting from the left of the arrow, the classifier or the suffix must be removed from the word to obtain the root, on the right is the result obtained after passing through the analyzer. Each rule of the algorithm is preceded by a condition indicating whether or not the suffixing rule can be applied.

For example: (m>0) classifier (degree 1) → ɛ.

"*palanɗe*"; we will have (m=2)ɗe → ɛ and we obtain the stem "*palan*".
"*faɓɓitijaŋngo*"; we will have (m=4)go → ɛ and we obtain the stem "*faɓɓitijaŋn*".
"m" can be 5 or 6, or even more. It depends on the vowel sequence (V) and the consonant sequence (C) in the word.

5.2.3 The Development Process of Stemming
Our algorithm processes a dictionary of 5186 words obtained from the dictionary of North Cameroon Noye Dominique (Noye 1989).The deletion of the classifiers and the consonantic variation are done based on transducers.

The flowchart in Figure 4 illustrates the steps of our stemming algorithm. Being an adaptation of Porter's algorithm, we have used steps 1 and 4 of this algorithm and have removed a number of steps that do not correspond to the morphological aspect of the Fulani language. We have thus integrated new steps 2, 3 and 5 to obtain the stem of the words.



.

FIGURE 4: Steps of the Adapted Porter's Algorithm to the Fulani Language.

Start: enter the word in the algorithm
Step 1: test the word in plural
Step 2: if the word in plural, transform the first consonant of the word into its singular
Step 3: if the word in singular or plural, delete the singular or plural classifier from the word
Step 4: if the resulting word ends in a vowel, delete the vowel
Step 5: generate the dictionary stem-word
End: end of algorithm

```
 //Loop while we have word in file
Whileno end filedo
      word ← file.readline()
      ifword is pluralthen
            Transform first consonant into its singular
            has_classifier ← true
            whilehas_classifierdo:
                  // get word classifier
                  classifier ← word.getClassifier()
                  // swicth into each case
                  switch (classifier):
                        caseis 2nd degree plural classifierthen
                              DeleteListOfSecondDegreePluralClassifier()
                        caseis 3rddegre plural classifierthen
                              DeleteLlistOfThirdDegreePluralClassifier()
                        caseis 4th degree plural classifierthen
                              DeleteListOfFourthDegreePluralClassifier()
                        default:
                              has_classifier ← false
                  endswitch
            endwhile
      else
            has_classifier ← true
            whilehas_classifierdo:
                  //get word classifier
                  classifier ← word.getClassifier()
                  //switch into each case
                  switch (classifier):
                        caseis 1st degree singular classifierthen
                              DeleteListOfFirstDegreeSingularClassifier()

                        caseis 2nd degree singular classifierthen
                              DeleteListOfSecondDegreeSingularClassifier()
                        caseis 3rddegre singular classifierthen
                              DeleteListOfThirdDegreeSingularClassifier()
                        caseis 4th degree singular classifierthen
                              DeleteListOfFourthDegreeSingularClassifier()
                        default:
                              has_classifier← false
                  endswitch
            endwhile
      endif
      Delete long voyel
      Write stem to new stem's file
endwhile
```

The pseudocode of the Porter algorithm applied to the Fulani language has the following structure After applying Porter's algorithm on 5186 Fulani words, we obtained 3484 distinct stems, that is a reduction of the number of words to 67.18%.

6. EVALUATION OF THE CHOSEN STEMMING ALGORITHMS FOR THE PEUL LANGUAGE

The evaluation of stemming algorithms is based on the calculation of precision and recall measures that are used especially in the context of information retrieval. We will thus use Christ Paice's method to evaluate our stemming algorithms. Nevertheless, these measures do not show

the specific causes of errors, which does not help designers to optimize their algorithms. Thus, Christ Paice (Paice 1994) proposed an evaluation method by introducing two measures: under stemming, which occurs when two related terms are not reduced to the same stem, and over stemming, which occurs when two unrelated terms are associated to the same stem by error. For this method, we need a sample of **w** words classified in **g** groups of concepts, whose $\mathbf{n_g}$ attributes in each group are morphologically and semantically related to each other. Thus, the under and over stemming error rates are calculated according to the following formulas, respectively:

$$\mathbf{US} = \frac{GUMT}{GDMT} \quad \text{and} \quad \mathbf{OS} = \frac{GWMT}{GDNT}$$

Where $GDMT = \sum_g \sum_{n_g} 0.5\, n_g\, (n_g - 1)$; $GDNT = \sum_g \sum_{n_g} 0.5\, n_g\, (W - n_g)$;
$GUMT = \sum_g \sum_{s_g} 0.5\, u_{S_g}\, (n_g - u_{s_g})$; and $GWMT = \sum_s \sum_{t_s} 0.5\, v_{t_s}\, (n_s - v_{t_s})$.

where **s** is the total number of distinct stem, s_g is the number of stem in a group, t_s and n_s are respectively the number of concepts and words reduced to the same stem, u_{s_g} the number of words reduced to the same stem in the group **g**, and v_{t_s} is the number of words reduced to the same stem and belonging to the concept **t.**

To evaluate the different approaches of stemming, we used a sample W =5186 different words, divided into 13 groups of concepts: animals, things, food, education, place, disease, profession, nature, person, quantity, religion, time, plants. Each containing forms, which are both semantically and morphologically, related to one another.

6.1 Result

After evaluating the Lovins and Porter stemming algorithms adapted to the Fulani language while using the Christ Paice method to find the error rate of under-stemming and over-stemming, we obtained the results below.

	Lovins	Porter
OS	0.00047	0.00044
US	1.01376	1.01382

TABLE 2: Evaluation results of the Porter and Lovins algorithms.

From the table above summarizing the evaluative results of the stemming algorithms adapted to the Fulani language, we see that both Lovins and Porter stemming algorithms are good for the Fulani language; the values of US are higher than OS, i.e. we have less over stemming errors. Christ Paice mentioned that when US>OS we are dealing with a lightweight algorithm. the under stemming error is mainly due to consonantal variation when some words change from singular to plural, so words belonging to the same family end up with different stem.

However, the results of the two algorithms are similar but to judge the performance, we will calculate ERRT of Lovins and ERRT of Porter always in the same logic as Christ Paice. Errt evaluates the performance of the algorithm with respect to the limit method, which is the truncation. To have a good stemming algorithm, P must belong to the segment [OT] and the point P and T must be very close.

$$\text{ERRT} = \frac{Lenght\ (OP)}{Lenght(OT)}$$

After the calculations, we obtained:

ERRT Lovins = 0.999765768905117
ERRT Porter = 0.999826455475208

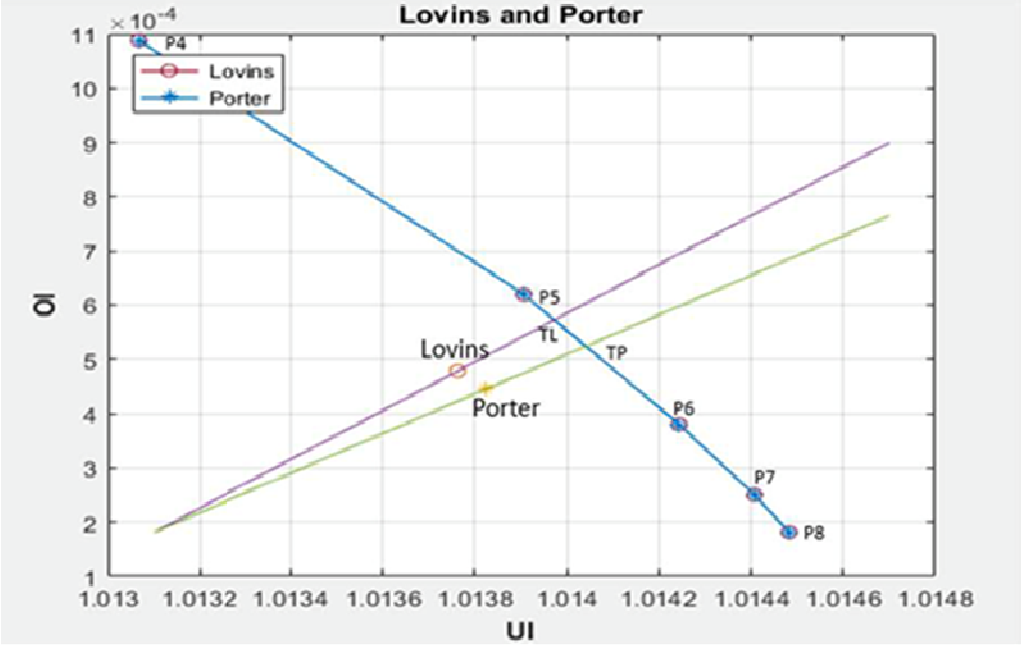


FIGURE 5: Computation of ERRT Value.

If we take ERRT as a general indicator of performance accuracy, we would have to conclude that Lovins is a better stemmer than Porter, however for the Fulani language, we can use one of the two algorithms for the stemming of words because the results are similar.

7. IMPACT AND SIGNIFICANCE OF THE STUDY

In this article, we have developed the FulLovins and FulPorter stemming algorithms by adapting the Lovins and Porter stemming algorithms to the Fulani language. To do this, we made a morphological study of Fulani words to obtain the stemming rules. We found that Fulani words do not have prefixes but only suffixes and some words change their initial consonants when they go from singular to plural and vice versa. For the removal of suffixes, we used transducers instead and the evaluation of these algorithms was done by the method of Christ Paice in order to see among the two algorithms which one is more efficient for the Fulani language. These algorithms allowed us to group semantically close words based on similarities in order to be used in NLP applications such as information retrieval systems, text summarization, machine translation, text categorization and morphological analysis tools. For a corpus of 5186 words, our Porter algorithm adapted to the Peul language reduced the corpus to 3484 distinct stems, i.e. a reduction of the corpus of 67.18%. As for Lovins adapted to the Peul language, still for a corpus of 5186 words, we obtained 3180 distinct stems, that is to say a reduction rate of the words of the corpus of 61.32%. Comparing to other languages, (Allah et al. 2010) have realized a tool for the pseudo-stemming of the Amazigh language, for a sample of 2171 distinct words. By applying their algorithm to this sample, they have established 1015 pseudo-stems, which allowed us to reduce the size of the elaborated sample by 53.2%. Moreover, using Paice's measures for the calculation of over- and under-stemming errors, they arrived at an error rate of 33.7% for under-stemming and 0.4% for over-stemming. In the same vein, (Samuel et al. 2018) designed a stemming algorithm for the Kambaata language. Using a corpus of 2425 distinct words, the evaluation results showed that 2349 words were stemming or 96.87%., the stemming algorithm thus reduced the corpus size by 65.86%.

We note from these results that the stemming algorithms designed for the Peul language are efficient. In addition, a word in Peul has quite important variants and the fusion of all these variants increases the performance of the research; it also decreases the storage space necessary for the documents of indexing.

8. CONCLUSION AND PERSPECTIVES

This article reviews the main algorithms of suffix stripping for natural language processing, with a focus on African languages. The focus is on the adaptation of Porter and Lovins stemming algorithms to the Peul language. We proceeded to a morphological analysis of the Peul language in order to adapt both Porter and Lovins algorithms which allow the stemming of words in Peul. By working on a dictionary containing 5186 words, we were able to reduce the corpus size from 5186 words to 3180 distinct words with Lovins and from 5186 words to 3484 distinct words with Porter. In addition, transducers were designed to help understand the different processing steps of these algorithms. In order to verify which of the two stemming algorithms is more efficient for the stemming of the Peul language, we have evaluated the performance indices of these algorithms using the Christ Paice method. The results of the truncation error rate or ERRT being approximately equal for Lovins and Porter, we can conclude that although ERRT Lovins is smaller than ERRT Porter, the results remain similar and that both stemming algorithms are efficient for the Fulani language. The present work is an incremental strategy to encourage automatic processing of the Fulani language through the development of basic tools and linguistic resources such as automatic summaries, text classification, document categorization, text translators...In the process of information retrieval, stemming algorithms will help linguists to improve the search process by its ability to index documents according to topics, and to broaden a search to obtain more accurate results.In order to improve this work, we need to extend the approach by integrating a module on the consonantal alternation feature of the letters W and Y, to consider compound words in Fulani and the management of common classifiers for singular and plural words. . Then, we will adapt the other stemming algorithms to the Fulani language and calculate the performance of the results.

9. REFERENCES

Al-Kharashi, I. A., & Evens, M. W. (1994). Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science*, *45*(8), 548-560.

Amidou, M. (2009). Bi-grammaire fulfulde/pulaar-français. Direction de l'Education et de la Formation : Programme d'apprentissage du français en contexte multilingue.

Arnott, D. W. (1960). *The tense system in Gombe Fula*. University of London, School of Oriental and African Studies (United Kingdom).

Ataa-Allah, F., & Boulaknadel, S. (2010, July). Pseudo-racinisation de la langue amazighe. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts* (pp. 44-49).

Boukhari, K. (2013). Un Nouvel Algorithme de Stemmatisation pour l'Indexation Automatique de Documents non-structurés : Stemmer SAID.

Cefan. Répartition du peul d'Afrique, URL : http://www.axl.cefan.ulaval.ca/afrique/peuls-map.htm, visited on 13-04-2021.

Conjugaison. Les formes verbales URL:http://www.conjugaison.com/grammaire/formes verbales.html, visited on 21-01-2021.

Darwish, K. (2002). *Building a shallow Arabic morphological analyzer in one day*.

Diallo, A. (2015). *Précis de grammaire et de lexique du peul du FoutaDjallon*. Research Institute for Languages and Cultures of Asia and Africa (ILCAA). Tokyo University of ForeignStudies.

Francois, Y. (2007). Transducteurs finis en Traitement des Langues. École Nationale Supérieure des télécommunications, Département Informatique et Réseaux, Paris.

Harrathi, F., Roussey, C., Calabretto, S., Maisonnasse, L., &Gammoudi, M. M. (2009). Indexation sémantique des documents multilingues. *INFORSID, editor, Atelier RISE associé au 27ème Congrès INFORSID*, 31-50.

Heine, B., & Nurse, D. (Eds.). (2000). *African languages: An introduction*. Cambridge University Press.

Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6),1930-1938.

Kevers, L., Gueniot, F., Tognotti, A. G., &Medori, S. R. (2019). Outiller une langue peu dotée grâce au TALN: l'exemple du corse et BDLC. In *26e Conférence sur le Traitement Automatique des Langues Naturelles* (pp. 371-380). ATALA.

Le, N. T. (2019). Traduction automatique pour une paire de langues peu dotée. These du Doctorat en Informatique Cognitive.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, *11*(1-2), 22-31.

Mahyoob, M. (2018). Deterministic Finite State Automaton of Arabic Verb System: A Morphological Study. *International Journal of Computational Linguistics (IJCL)*, *9*(1).

Majumder, P., Mitra, M., &Datta, K. (2006, September). Statistical vs. rule-based stemming for monolingual french retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages* (pp. 107-110). Springer, Berlin, Heidelberg.

Mohamadou, A. (1991). *Classificateurs et représentation des propriétés lexicales en peul: parlers de l'Aadamaawa* (Doctoral dissertation, Paris, INALCO).

Mohamadou, A. (2014). *Le verbe en peul: Formes et valeurs en pulaar du Fuuta-Tooro*. KARTHALA Editions.

Noye, D. (1974). Cours de foulfouldé: dialecte peul du Diamare, Nord-Cameroun.

Noye, D. (1989). *Dictionnaire foulfouldé-français: dialecte peul du Diamaré, Nord-Cameroun*. Librairie Orientaliste Paul Geuthner.

Omri, M. N., &Chouigui, N. (2001). Linguistic variables definition by membership function and measure of similarity. In *Proceedings of the14th International Conference on Systems Science* (Vol. 2, pp. 264-273).

Omri, M. N. (2004). Possibilistic pertinence feedback and semantic networks for goal extraction. *Asian Journal of Information Technology*, *3*(4), 258-265.

Paice, C. D. (1994). An evaluation method for stemming algorithms. In *SIGIR'94* (pp. 42-50). Springer, London.

Paternostre, M., Francq, P., Lamoral, J., Wartel, D., & Saerens, M. (2002). Carry, un algorithme de désuffixation pour le français. *Rapport technique du projet Galilei*.

Porter, M. F. (1997). An algorithm for suffix stripping. Readings in information retrieval. *Morgan Kaufmann*, 313-316.

Samuel, J., Teferra, S., Samuel, J., Teferra, S., Samuel, J., &Teferra, S. (2018). Designing A Rule Based Stemming Algorithm for Kambaata Language Text. *no*, *9*, 41-54.

Taylor, F. W. (1953). A grammar of the Adamawa dialect of the Fulani language (Fulfulde).

Tesfaye, D., & Abebe, E. (2010). Designing a Rule Based Stemmer for Afaan Oromo Text. *International journal of computational linguistics (IJCL)*, *1*(2), 1-11.

Tradlibre. Histoire de la langue Peuls. URl: https://www.tradlibre.fr/histoire/histoire-de-la-langue-peuls, visited on 7-04-2021

Younoussi, Y. E., Sdigui, A.D.,Belahmer, H. (2007). La racinisation de la langue arabe par les automates à états finis (AEF). Laboratoire Systèmes d'Information Multimédia et Mobiles (SI3M), Ecole Nationale Supérieure de l'Informatique et Analyse des Systèmes Maroc, Laboratoire Alkhawarizmi de Génie Informatique (LAGI).