Journal of Computer Science

Original Research Paper

A Generic Syntax Analyzer for the Computerization of African Languages

Yahaya Morou Ganda, Harouna Naroua and Bachir Moussa Idi

Département de Mathématiques et Informatique, Faculté des Sciences et Techniques, Université Abdou Moumouni, Niamey, Niger

Article history
Received: 13-08-2024
Revised: 03-12-2024
Accepted: 26-12-2024

Corresponding Author:
Yahaya Morou Ganda
Département de Mathématiques et Informatique, Faculté des Sciences et Techniques, Université Abdou Moumouni, Niamey, Niger
Email: y.morouganda.35116@gmail.com

Abstract: This article enters within the framework of a contribution to the computerization of African Languages. It presents a generic parsing tool designed for West African languages despite their individual specificities. The genericity of the tool resides in the fact that it is adaptable to any West African language while using a minimum of resources. It only needs an etiquette dictionary and context-free grammar. From these two resources, the tool is designed to generate an LR (1) parser. The choice of this technique is based on its supremacy over other techniques despite its disadvantage of generating an exponentially large parsing table. However, to reduce the size of the automaton while maintaining the power of the technique, the system brings together dictionary entries depending on their type, gender, and number in the form of terminals. The results obtained show that it is possible to generate a relatively small and efficient analysis table instead of a large one. An application of the proposed tool to the Hausa nominal phrases has succeeded in reducing the number of terminals from more than 10,000 to only 88. Instead of having an automaton with thousands of states, we end up with an automaton with only 198 states.

Keywords: Natural Language Processing, Syntax Analysis, Automaton, Generic Tool, African Language

Introduction

Started in the United States in the 1950s (Fuchs and Habret, 2004), Natural Language Processing (NLP) aims to create natural language processing tools for various applications. It is a field that brings together several disciplines such as linguistics, computer science, and artificial intelligence. NLP is especially used in the study of the morphology, syntax, or semantics of a language. It uses several methods including neural networks and automata. The latter are defined by Abdullahi (2021) as abstract machines that know how to recognize the appurtenance or non-appurtenance of a word to a given regular expression. They are indispensable means in word processing and many other domains including Compilation, electronics, model-checking, etc.

From constituent parsing (Floquet *et al.*, 2023) to parsing based on artificial intelligence (Zhao and Bernard, 2023), passing through statistical approaches (Regnault, 2022), several methods are used to facilitate the processing of highly endowed languages. Unlike other methods, automata-based methods do not require a very large amount of data to provide excellent results. This peculiarity makes automata an ideal way to facilitate the processing of poorly endowed languages, including African languages considered to be languages that are mostly spoken but rarely documented (Konan Groguhe, 2024) such as Hausa.

Much more than just a means of communication, language is a vibrant reflection of a people's cultural diversity, identity and history. By understanding and preserving the richness of spoken languages, we open ourselves to multiple perspectives, ancestral knowledge and deep bond that unite communities. For Costa (2021), languages contribute to the total knowledge of humanity and linguistic diversity is essential to life on earth.

According to the 20th edition of the Ethnologue (https://www.ethnologue.com/) of 2017, our planet has 7100 languages spread over more than 220 countries (or states). According to UNESCO, about half of these languages are in danger of extinction in the long term. In the short term, it is estimated that one language disappears every two weeks, which equates to about 26 languages per year.

In Africa, the situation is even more alarming. Indeed, the number of African languages is estimated at about 2000 (1⁄3 of the world's languages). But only about 400

Science Publications

© 2025 Yahaya Morou Ganda, Harouna Naroua and Bachir Moussa IDI. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

have been described. Nearly 1,600 languages have not been seriously studied and remain under threat of possible extinction. While all of these languages are in principle capable of expressing everything, (Frath and Sow, 2022) believe that only a few of them are currently able to provide access to the knowledge of modernity. For example, despite its large number of speakers (72 million), which makes it the second most spoken language in Africa after Swahili, the Hausa language is not on the sidelines of this threat.

Thus, in order to guarantee the survival of African languages and to allow their emergence in the digital world, it would be opportune to find an effective way to facilitate their automatic processing.

If the importance of automata is no longer to be demonstrated in syntactic analysis, with the existence of several word processing systems for several languages, it is clear that African languages have not benefited from the advantages they offer. Thus, to fill in this gap and allow African languages to be brought up to the same level as other languages in the field of word processing, this study aims to develop a generic tool that will automatically generate syntax analyzers for West African languages with application to the nominal phrases of one of the most widely spoken languages in Africa, namely Hausa. The proposed tool is a computer program based on algorithms that are not only able to adapt to the peculiarities of a particular language like Hausa whose syntactic structure is very complex, but also this program will need only a minimum of resources such as a dictionary and a descriptive grammar of the language to provide an effective parser.

Brief Overview of Parsing Techniques

According to the Oxford Dictionary, syntax is the set of rules and principles in a language according to which words, phrases, and clauses are arranged to create well-formed sentences. It allows us to reveal the grammatical rules of a language and to understand how ideas are expressed. Syntactic analysis is defined by Grune and Jacobs (1990) as "the process of structuring a linear representation in accordance with a given grammar". It is a field of computer science that was extensively studied during the 1950s, even before the theory of formal languages developed by Chomsky (1956). In the decade that followed, several techniques based on this theory emerged. Thus, John Cocke initiated a dynamic programming algorithm in 1960 which he did not formally publish. His algorithm was later formalized by Kasami (1966); Younger (1967) and constituted the famous CKY algorithm, analyzing a sequence in cubic time, using Chomsky's normal-form grammars. In 1968, Jay Earley invented the first cubic time parsing algorithm for all context-free grammars. During the 1970 s and 1980s, several other methods emerged, including Augmented Transition Networks (ATN) and Constraint-Based Grammar. In the same period, the theoretical principles of syntactic analysis were written by renowned scientists such as (Aho *et al*. 1986; Knuth, 1965) to name but a few. Parsing algorithms can be classified into several categories characterized by the strategies they use.

Tabular Algorithms

Mainly consisting of CKY and Earley's algorithms, tabular algorithms use dynamic programming techniques that make them particularly well-suited for processing ambiguous grammars. The CKY algorithm operates by bottom-up analysis and assumes that the grammar is in Chomsky's normal form. As for Earley's, it is a general parsing algorithm developed by Jay Earley (1970) as part of his doctoral thesis. The paper on the technique was first published in the journal "Communications of the ACM" under the title "An efficient context-free parsing algorithm". It takes as input a context-free grammar and a list of words representing the sentence. The algorithm is based on four main functions which are: The scan function, the predict function, the complete function, and the recognize function which is the entry of the algorithm (Ouellette-Vachon, 2013). Tabular algorithms include Bottom-up Chart Parsing (Nakamura-Delloye, 2003), which is an algorithm for bottom-up and parallel parsing. It is characterized by the fact that it allows the rule of generating an empty string and does not require the use of grammar in normal Chomsky form.

Deterministic Algorithms

Deterministic methods are mainly made up of top-down and bottom-up algorithms. Examples of top-down methods include predictive analytics and recursive descent. Predictive analytics uses a production table whose rows are denoted by the non-terminal symbols of the grammar and the columns by the terminal symbols and the end symbol of the string. To construct this table, the boxes that may contain grammar rules, and the next and first functions must be determined (Aho *et al*., 1986). One of the most popular predictive analytics algorithms is the LL algorithm. For the recursive method, it is considered to be slightly naïve, as it tests all symbols until one of them works.

The principle of bottom-up analysis is to construct a bottom-up derivation tree using the offset/reduction principle. LR analysis is the most well-known of the bottom-up methods and allows the analysis of context-independent grammars, also known as LR (k) grammars. This method uses a table that can be constructed in three different ways (Aho *et al*., 1986):

- Simple LR (SLR): This is the simplest to implement, but the least powerful of the three

DOI: 10.3844/jcssp.2025.810.816

- Canonical LR: Which is the most powerful, but also the most expensive
- Look Ahead LR (LALR): which has intermediate power and cost between the other two and can be applied to the majority of programming language grammar

Materials and Methods

The Methods Used

Among the multitude of methods that exist for syntax analysis of a language, canonical LR parsing was chosen. In fact, several studies (Aho *et al*., 1986) have shown the supremacy of bottom-up methods over top-down methods. Studies have also shown that LR (1) analysis is the most powerful bottom-up method and, hence the most powerful deterministic algorithm (Aho *et al*. (1986). It works with the shift/reduction principle and uses a pushdown automaton whose states are constructed from a collection of item sets as described by Nakamura-Delloye (2003).

System Operation

The system is developed in Java under the NetBeans 14 integrated development environment. It is made up of a set of classes where the principal ones are shown in Fig. (1).

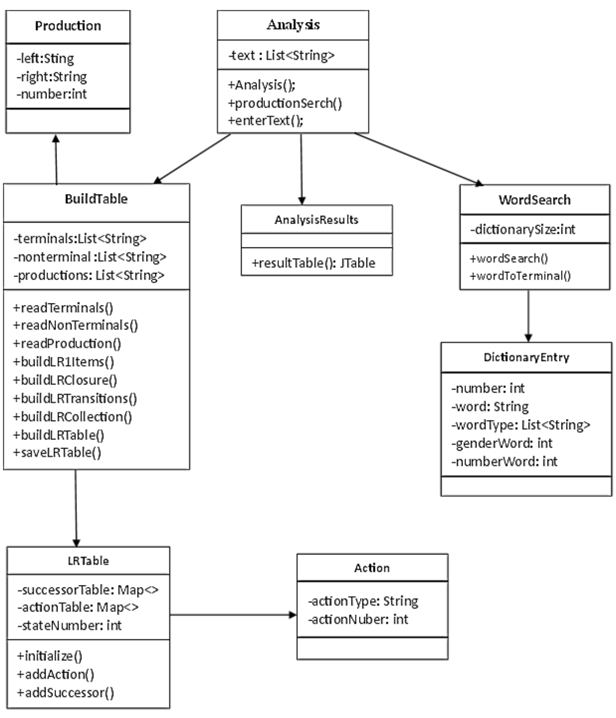


Fig 1: System class diagram

To perform an analysis, it is possible to construct a new table or use an existing table. The construction of a table, consisting of an ACTION part and a SUCCESSOR part, is done from the Build table class by reading as input the terminals and non-terminals as well as the production rules. The states of the automaton are constructed from a COLLECTION of sets of items where each set represents a state. To build this collection, the CLOSURE and TRANSITION sets are to be determined by using the build LR Closure () and build LR Transitions () functions respectively. To reduce the size of the automaton, the proposed system groups dictionary words according to their types, genders, and numbers. A transformation function has been implemented for that purpose.

The text to be analyzed is broken down into words and placed in a list. When the reading head is on a word, it is searched in the dictionary (Word Search class) and if it is found then it is reduced to its type with a combination of its gender and number. Semantically, the sense of a sentence is considered to be the combination of the senses of the words that make it up. A plausible combination indicates that the sentence is syntactically correct. If no plausible combination is found, then the sentence is syntactically incorrect. Also, if the word is not found in the dictionary, then it is considered not to belong to the considered language. The Analysis Results class is used to display the results of the analysis in a tabular form.

Application to the Hausa Nominal Phrases

Spoken in several African countries, mainly in West Africa and more particularly in Niger and Nigeria, the Hausa language belongs to the Afro-Asiatic language family and is classified in the Chadic subgroup like Berber, Arabic and Tamajag. It is the most widely spoken language in the Chadian branch and the second most widely spoken language in Africa after Swahili. Ethnologists have estimated that it is spoken as a first language by some 47 million people and as a second language by another 25 million, bringing the total number of its speakers to about 72 million. In Niger, Hausa is spoken by nearly 55% of the population and has eight dialects, depending on the area of influence, namely: Damagaranci, Katsinanci, Gobiranci, Adaranci, Arewanci, Kurfayanci, Agadasanci and Dandanci. Regarding its phonetics, written Hausa is essentially based on the Kano dialect and has two writing systems: Ajami which uses Arabic alphabets, and Boko which uses the Latin alphabet Fig. (2) (Abdullahi, 2021).

Hausa has two grammatical genders which are Masculine (M) and Feminine (F). In most cases, the feminine can be formed from the masculine. Generally, nouns and adjectives that end with "a" are feminine. Hausa sentences are non-verbal or verbal. Verbal sentences are simple or complex. For non-verbal or nominal sentences, the actualization of the

DOI: 10.3844/jcssp.2025.810.816

statement is not ensured by a verb but by one of the six predicative particles which are: Ne/ce "it is" and its negation ba. Ba "it's not"; the existential akwai "there is" and the negative equivalent babu/ba "there is not"; The presentative ga "there is/are" and the directional particle za "go in the direction of" (Caron, 2011). Nominal sentences can be broken down into three parts, which are the header optionally followed by a post-header and optionally preceded by a pre-header. The header can be a noun or a pronoun and the pre-headers and post-headers can be adjectives among others.

The development of an automatic language processing tool is done in three stages, the overall order of which cannot be reversed: Resources, computer tools, and applications (Jouitteau, 2023). Dictionaries are essential linguistic resources, particularly in spell correction, and are very valuable for the proposed system because they will considerably reduce the number of grammar productions that will be used. Several dictionaries exist for the Hausa language, including the Hausa-French dictionary with 7823 entries from the Soutéba project (support program for basic education) (Enguehard *et al*., 2011), the Hausa-French dictionary of (Adamou, 2014) 7238 entries and the Hausa-French dictionary of the Nigerien linguist Abdou (2003) used by Naroua and Salifou (2016). The latter, with 10,000 entries and whose author is himself a Hausa native, is the one chosen for the proposed syntax analyzer of nominal phrases because of its largest number of entries. Table (1) shows an extract from the dictionary in a format adapted to the system in which the following is used:

Table 1: Extract from the Hausa dictionary

Number	Word	Type	Gender	Number
0	a	v_n_adj	0	0
1	a'a	*	9	9
2	a-gashe	n	1	1
3	abada	n	0	1
4	abaici	n	0	1
5	abara	n	0	1
6	abarba	n	0	1
7	abawa	n	0	1
8	abba	n	1	1
9	abduga	n	0	1
10	abinci	n	1	1
11	abkawa	*	9	9
12	abki	n	1	1
13	abkuwa	*	9	9
187	alhali	adv	1	1
363	arzitacce	adj	1	1
400	aski	n_v	1	1
2014	gaman	conj	9	9
2024	game	n_v_co nj	1	1
2327	goma	card	9	9

Gender:	Number:
0: Feminine	0: Plural
1: Masculine	1: Singular
9: Undefined	9: Undefined

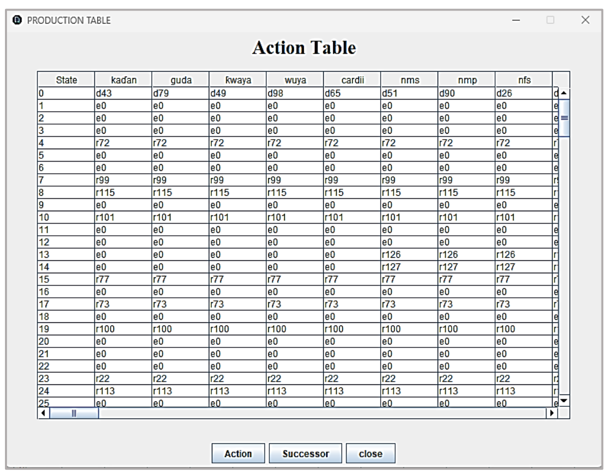
Type:

- N: Noun
- V: Verb
- Adj: Adjective
- adv: Adverb
- it: Interjection
- IPA: Indicator of Person/aspect/time
- conj: Conjunction
- excl: Exclamation
- card: Cardinal or Numeral
- Ord: Ordinal
- neg: Negation
- St: Stand
- prep: Preposition
- Id: Ideophone
- dem: Demonstrative
- appr: Appreciative
- poss: Possessive
- expr: Expression / Expressive
- deer: Depreciative
- itg: Interrogation
- dim: Diminutive
- pr: Pronoun
- aux: Auxiliary
- epars: Scattered
- prpers: Personal Pronoun
- det: Determinative
- ind: Indicative
- ant: Antonyme
- reit: Reiteration / Reiterative
- rel: Relative
- * : Indefinite

A context-free grammar composed of 88 terminals, 27 nonterminals, and 129 productions, was constructed in place of a grammar that was supposed to have thousands of terminals and productions.

Results

Figures (3-9) show extracts of the ACTION and SUCCESSOR parts of the transition table and the results of analyses carried out on some Hausa nominal phrases.



State	kadan	guda	kwaya	wuya	cardii	nms	nmp	nfs
0	d43	d79	d49	d98	d65	d51	d90	d26
1	e0	e0	e0	e0	e0	e0	e0	e0
2	e0	e0	e0	e0	e0	e0	e0	e0
3	e0	e0	e0	e0	e0	e0	e0	e0
4	r72	r72	r72	r72	r72	r72	r72	r72
5	e0	e0	e0	e0	e0	e0	e0	e0
6	e0	e0	e0	e0	e0	e0	e0	e0
7	r99	r99	r99	r99	r99	r99	r99	r99
8	r115	r115	r115	r115	r115	r115	r115	r115
9	e0	e0	e0	e0	e0	e0	e0	e0
10	r101	r101	r101	r101	r101	r101	r101	r101
11	e0	e0	e0	e0	e0	e0	e0	e0
12	e0	e0	e0	e0	e0	e0	e0	e0
13	e0	e0	e0	e0	e0	r126	r126	r126
14	e0	e0	e0	e0	e0	r127	r127	r127
15	r77	r77	r77	r77	r77	r77	r77	r77
16	e0	e0	e0	e0	e0	e0	e0	e0
17	r73	r73	r73	r73	r73	r73	r73	r73
18	e0	e0	e0	e0	e0	e0	e0	e0
19	r100	r100	r100	r100	r100	r100	r100	r100
20	e0	e0	e0	e0	e0	e0	e0	e0
21	e0	e0	e0	e0	e0	e0	e0	e0
22	e0	e0	e0	e0	e0	e0	e0	e0
23	r22	r22	r22	r22	r22	r22	r22	r22
24	r113	r113	r113	r113	r113	r113	r113	r113
25	e0	e0	e0	e0	e0	e0	e0	e0

Fig. 3: Extract from the ACTION part of the transition table

DOI: 10.3844/jcssp.2025.810.816

PRODUCTION TABLE

Successor Table

State	START	NP	HD	PRE_HD	POST_HD	ADJ_PH	QT	EN	
0	-1	70	12	74	-1	30	62	89	2
1	-1	-1	-1	-1	-1	-1	-1	-1	-
2	-1	-1	-1	-1	-1	-1	-1	-1	-
3	-1	-1	-1	-1	-1	-1	-1	-1	-
4	-1	-1	-1	-1	-1	-1	-1	-1	-
5	-1	-1	-1	-1	-1	-1	-1	-1	-
6	-1	-1	-1	-1	-1	-1	-1	-1	-
7	-1	-1	-1	-1	-1	-1	-1	-1	-
8	-1	-1	-1	-1	-1	-1	-1	-1	-
9	-1	-1	-1	-1	-1	-1	-1	-1	-
10	-1	-1	-1	-1	-1	-1	-1	-1	-
11	-1	-1	-1	-1	-1	-1	-1	-1	-
12	-1	-1	-1	-1	162	-1	-1	-1	-
13	-1	-1	-1	-1	-1	-1	-1	-1	-
14	-1	-1	-1	-1	-1	-1	-1	-1	-
15	-1	-1	-1	-1	-1	-1	-1	-1	-
16	-1	-1	-1	-1	-1	-1	-1	-1	-
17	-1	-1	-1	-1	-1	-1	-1	-1	-
18	-1	-1	-1	-1	-1	-1	-1	-1	-
19	-1	-1	-1	-1	-1	-1	-1	-1	-
20	-1	-1	-1	-1	-1	-1	-1	-1	-
21	-1	-1	-1	-1	-1	-1	-1	-1	-
22	-1	-1	-1	-1	-1	-1	-1	-1	-
23	-1	-1	-1	-1	-1	-1	-1	-1	-
24	-1	-1	-1	-1	-1	-1	-1	-1	-
25	-1	-1	-1	-1	-1	-1	-1	-1	-

Action Successor close

Fig. 4: Extract from the SUCCESSOR part of the transition table

Analysis result

remaining string	word being analyzed	the possible types	type under test	exit
wani baƙo mai littahi babba $	wani	[specidftms, hd]	specidftms	shift : 88
baƙo mai littahi babba $	baƙo	[nms]	nms	reduce : SPEC-->[specidftms]
baƙo mai littahi babba $	baƙo	[nms]	nms	reduce : PRE_HD-->[SPEC]
baƙo mai littahi babba $	baƙo	[nms]	nms	shift : 51
mai littahi babba $	mai	[agcp]	agcp	reduce : N-->[nms]
mai littahi babba $	mai	[agcp]	agcp	reduce : HD-->[N]
mai littahi babba $	mai	[agcp]	agcp	shift : 24
littahi babba $	littahi	[nms]	nms	reduce : AG_CP-->[agcp]
littahi babba $	littahi	[nms]	nms	shift : 51
babba $	babba	[adjfs]	adjfs	reduce : N-->[nms]
babba $	babba	[adjfs]	adjfs	reduce : HD-->[N]
babba $	babba	[adjfs]	adjfs	shift : 160
$	$	[$]	$	reduce : UNIT_ADJ-->[adjfs]
$	$	[$]	$	reduce : POST_HD-->[UNIT_ADJ]
$	$	[$]	$	reduce : NP-->[HD, POST_HD]
$	$	[$]	$	reduce : ADJ_CX-->[AG_CP, NP]
$	$	[$]	$	reduce : POST_HD-->[ADJ_CX]
$	$	[$]	$	reduce : NP-->[PRE_HD, HD, POST_HD]
$	$	[$]	$	Conclusion : 'accepted'

close

Fig. 5: Analysis of the sentence wani baƙo mai littahi babba

Analysis result

remaining string	word being analyzed	the possible types	type under test	exit
waɗannan mutane masu alheri $	waɗannan	[adjms]	adjms	shift : 93
mutane masu alheri $	mutane	[nmp]	nmp	reduce : UNIT_ADJ-->[adjms]
mutane masu alheri $	mutane	[nmp]	nmp	reduce : PRE_HD-->[UNIT_ADJ]
mutane masu alheri $	mutane	[nmp]	nmp	shift : 90
masu alheri $	masu	[agcp]	agcp	reduce : N-->[nmp]
masu alheri $	masu	[agcp]	agcp	reduce : HD-->[N]
masu alheri $	masu	[agcp]	agcp	shift : 24
alheri $	alheri	[nms]	nms	reduce : AG_CP-->[agcp]
alheri $	alheri	[nms]	nms	shift : 51
$	$	[$]	$	reduce : N-->[nms]
$	$	[$]	$	reduce : HD-->[N]
$	$	[$]	$	reduce : NP-->[HD]
$	$	[$]	$	reduce : ADJ_CX-->[AG_CP, NP]
$	$	[$]	$	reduce : POST_HD-->[ADJ_CX]
$	$	[$]	$	reduce : NP-->[PRE_HD, HD, POST_HD]
$	$	[$]	$	Conclusion : 'accepted'

close

Fig. 6: Analysis of the sentence waɗannan mutane masu alheri

Analysis result

remaining string	word being analyzed	the possible types	type under test	exit
wani abu ƙidayayye alheri mai $	wani	[specidftms, hd]	specidftms	shift : 88
abu ƙidayayye alheri mai $	abu	[nms]	nms	reduce : SPEC-->[specidftms]
abu ƙidayayye alheri mai $	abu	[nms]	nms	reduce : PRE_HD-->[SPEC]
abu ƙidayayye alheri mai $	abu	[nms]	nms	shift : 51
ƙidayayye alheri mai $	ƙidayayye	[adjms]	adjms	reduce : N-->[nms]
ƙidayayye alheri mai $	ƙidayayye	[adjms]	adjms	reduce : HD-->[N]
ƙidayayye alheri mai $	ƙidayayye	[adjms]	adjms	shift : 124
alheri mai $	alheri	[nms]	nms	Conclusion : 'error'

close

Fig. 7: Analysis of the sentence wani abu ƙidayayye alheri mai

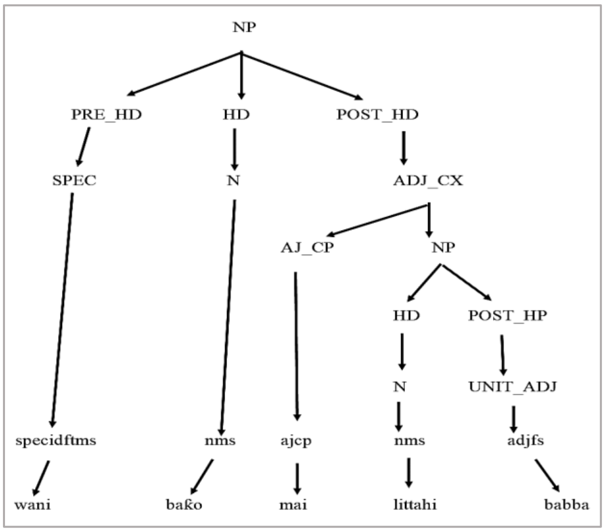


Fig. 8: Syntactic tree of the sentence wani baƙo mai littahi babba

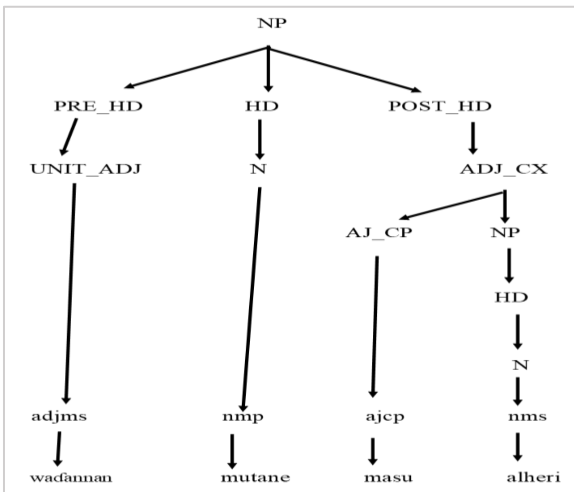


Fig. 9: Syntactic tree of the sentence waɗannan mutane masu alheri

DOI: 10.3844/jcssp.2025.810.816

Discussion

Despite its power, the canonical LR parser is known to have an automaton with a number of states that can grow exponentially. To reduce the size of the automaton, the proposed system groups dictionary words according to their type, gender, and number. This has significantly reduced the number of terminals from more than 10,000 to just 88 when the tool has been applied to Hausa nominal phrases. Instead of having thousands of states, we end up with a portable automaton. So, using a grammar of 88 terminals, 27 nonterminal, and 129 productions, a pushdown automaton of 198 states has been generated. Extracts of the ACTION and SUCCESSOR parts of the transition table are shown in Figs. (3-4).

Figures (5-7) show the results of analyses carried out on some Hausa nominal phrases.

On the result of the analysis of the first sentence, it is observed that the word "wani" has two types which are specidftms and hd. In this kind of situation, the program tests the different possible types for the word until it finds the type that will give the sentence the correct syntax. When it reaches the starting symbol at the end of the input string, then the sentence is accepted. Otherwise, it is rejected. Figures (8-9) represent the syntactic trees of the results in Figs. (5-6) respectively. They are constructed from the productions emitted as output each time a reduction is carried out.

Conclusion and Perspectives

Throughout this study, it has been noted, especially with the state of the art, that most of the methods that allow for an efficient syntax analysis for natural language are based on algorithms that require a very large amount of data on the language. This is a real obstacle for processing African languages, most of which have very little digital data. Nevertheless, it appears that automata-based methods produce very efficient results even with very little data and adapt well to African languages despite their specificities. Thus, with descriptive grammar and a dictionary, it has been possible to generate an efficient syntax analyzer for Hausa nominal phrases. Compared to the Bambara, Wolof and Yoruba syntax analyzers of Dione (2021) which require very large amount of data from three high-resource languages (English, French, Norwegian) and three low-resource languages (Bambara, Wolof and Yoruba), the proposed parser uses small amount of data from African languages only.

In perspective, it is planned to:

- Extend this program to the Hausa language as a whole by allowing the analysis of verbal and nominal sentences
- Extend this system to other West African languages
- Test and validate the proposed system in authentic contexts to ensure that it can function appropriately in various situations and with multiple types of data

Acknowledgment

We are thankful to Abdou Moumouni University of Niamey (NIGER) for facilitating the necessary support.

Funding Information

Research fund: Research grant from Abdou Moumouni University of Niamey (NIGER).

Author’s Contributions

Yahaya Morou Ganda: Made considerable contributions to design, acquisition of data, analysis and interpretation of data; Contributed in drafting the article and reviewing it critically for significant intellectual content; Prepare the final version to be submitted and any revised version.

Harouna Naroua: Coordinated the research activity and made considerable contributions to design, analysis and interpretation of data; Contributed in reviewing the article critically for significant intellectual content; Gave final approval of the version to be submitted and any revised version.

Bachir Moussa Idi: Made considerable contributions to design, analysis and interpretation of data; Contributed in reviewing the article critically for significant intellectual content; Gave final approval of the version to be submitted and any revised version.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

References

Abdou, M. (2003). *Dictionnaire Elémentaire Hausa-Français*.

Abdullahi, B. S. (2021). Aperçu des Système Phonologiques du Français et du Haoussa, RANEUF, Department of Linguistics and Foreign Languages. *RANEUF*, 18-32,

Adamou, M. R. (2014). *Dictionnaire Bilingue Haoussa-Français, Dictionnaires Langues Africaines-Français (DiLAF*.

Aho, A. V., Lam, M. S., Ullman, J. D., & Avaya, R. S. (1986). *Compilers: Principles, Techniques & Tools, Pearson Addison Wesley*.

Caron, B. (2011). *Haoussa, Dictionnaire des Langues, Presses Universitaires de France*. 263-269.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, *2*(3), 113-124.
https://doi.org/10.1109/TIT.1956.1056813

Costa, J. (2021). Langues en Danger. *Langage et Société*, *Hors série* (HS1), 201-204. https://doi.org/10.3917/ls.hs01.0202

Dione C. H. (2021), Multilingual Dependency Parsing for Low-Resource African Languages: Case Studies on Bambara, Wolof, and Yoruba, Proceedings of the 17th International Conference on Parsing Technologies Bangkok, Thailand 84–92.

Earley, J. (1970). An Efficient Context-Free Parsing Algorithm. *Communications of the ACM*, *13*(2), 94-102. https://doi.org/10.1145/362007.362035

Enguehard, C., Kané, S., Mangeot, M., Modi, I., & Sanogo, M. (2011). *Vers l'informatisation de quelques langues d'Afrique de l'Ouest"*. 13-32.

Floquet, N., Roux, J. L., Tomeh, N., & Charnois, T. (2023). Attention Sur Les Spans Pour l'analyse Syntaxique en Constituants. *Actes de La 30e Conférence Sur Le Traitement Automatique Des Langues Naturelles (TALN*, 37-45.

Frath, P., & Sow, N. (2022). La Question de La Terminologie Dans l'enseignement des Langues Africaines. *Linguistique Pour Le Développement. Concepts, Contextes et Empiries*, 97-114. https://doi.org/10.17184/eac.5250

Fuchs, C., & Habret, B. (2004). Le Traitement Automatique des Langues: Des Modèles Aux Ressources, Article Paru Dans Le Français Moderne LXXII. *Version Préliminaire*, 1-9.

Grune, D., & Jacobs, C. J. H. (1990). *Parsing Techniques: A Practical Guide. Computers and their Applications.*

Jouitteau, M. (2023). Guide de Survie Des Langues Minorisées à l'heure de l'intelligence Artificielle : Appel aux Communautés Parlantes. *Lapurdum*, *24*, 199-217. https://doi.org/10.4000/127u1

Kasami, T. (1966). An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Knuth, D. E. (1965). On the Translation of Languages from Left to Right. *Information and Control*, *8*(6), 607-639. https://doi.org/10.1016/s0019-9958(65)90426-2

Konan Groguhe, A. (2024). L'Intelligence Artificielle: De La Disparité Dans La Promotion des Langues. *Communication, Technologies et Développement*, *15*. https://doi.org/10.4000/123iv

Nakamura-Delloye, Y. (2003). *Analyse Syntaxique du Japonais, Mémoire de DEA, Institut National des Langues et Civilisations Orientales (INALCO*.

Naroua, H., & Salifou, L. (2016). On the Computerization of African Languages. *American Journal of Applied Sciences*, *13*(11), 1228-1234. https://doi.org/10.3844/ajassp.2016.1228.1234

Ouellette-Vachon, M. (2013). Méthodes de diagnostic d'erreurs d'analyse syntaxique (Doctoral dissertation, École Polytechnique de Montréal). *Université de Montréal*, 1–79.

Regnault, M. (2022). Annotation et analyse syntaxique de corpus hétérogènes: le cas du français médiéval. *Université de La Sorbonne Nouvelle - Paris III*.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n3. *Information and control*, *10*(2), 189-208.

Zhao, F., & Bernard, T. (2023). Auto-Apprentissage et Renforcement Pour une Analyse Jointe Sur Données Disjointes: Etiquetage Morpho-Syntaxique et Analyse Syntaxique. *Actes de La 30e Conférence Sur Le Traitement Automatique des Langues Naturelles (TALN*, 82-90.