

CERTIFICATION

This project with the title **development of an English to Yoruba machine translation system using direct based approach,**

Submitted by

ADEBAYO, PROMISE SOORE (CPE/13/1071)

Has certified the regulations governing the award of degree of

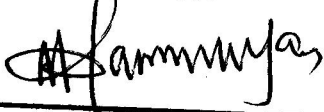
BACHELOR OF ENGINEERING (B.Eng) in COMPUTER ENGINEERING

Federal University, Oye-Ekiti, Ekiti.



Engr. Mrs. A. Esan

SUPERVISOR



Dr. Olaniyan

HEAD OF DEPARTMENT.

01-04-2019

Date

11/4/19

Date

DECLARATION

This project is a result of my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, internet etc.) has been acknowledged within the main report to an entry in the references list.

ADEBAYO PROMISE SOORE.

Student's full name



01/04/2019

Signature and date.

DEDICATION

I dedicate this project to the Almighty God – The Alpha and Omega of my life, my loving parents and siblings.

ACKNOWLEDGEMENT

My profound gratitude goes to God for his grace and sufficiency throughout the period of this project. I would also love to appreciate my parents, Pst. and Mrs., Paul Adebayo, for their unending support, I love you loads. Special thanks to Odole Kayode for being ever ready to answer my unending question. My heartfelt thanks to my siblings, Oluwadunsin, Erioluwa and Oluwaferanmi for supporting me and also to my friends and colleagues who has in one way or the other made this project journey easier.

I also acknowledge my project supervisor, Engr., Mrs. Esan and the head of department, Engr., O. Olaniyan, May God continue to bless you abundantly.

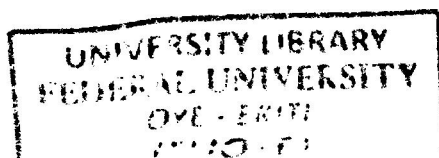
ABSTRACT

Language is the basic medium of communication and it is ultimately important because it is the primary means through which humans have the ability to communicate and interact with one another. Translation has been made very important so as to bridge the gap of information inequality. A lot of countries and organizations rely on human translators to disseminate information but human translators are limited in terms of speed, insecurity and high cost, therefore the demand for faster and cheaper means of translation. Hence, an English to Yoruba statistical machine translator was developed in this research.

The project was developed using python programming language and PyQt designer was used to design the graphic user interface. The direct method of translation was used to create a database of 25,000 words, which were gotten from locally spoken words and dictionaries. The system was evaluated using Mean opinion score (average) and tested using three different datasets.

The results from evaluation of the system based on percentage accuracy obtained from three different datasets are: 92.6%, 87.4%, 89.1% for dataset 1, 2 and 3 respectively. Thus, the percentage accuracy of the developed system was found to be 89.7% and was derived from the average percentage accuracy of the different datasets.

In conclusion, an English to Yoruba Translator was developed in this research to overcome the shortcomings of human translators and aid commercial activities. It is recommended that government should invest in machine translators to aid socio-economic growth of Yoruba-speaking states.



LIST OF ACRONYMS

MT – Machine translation

SMT – Statistical Machine Translation

DMT – Direct Machine Translation

RBMT – Rule based Machine Translation

EMBT – Example based Machine Translation

SL – Source Language

TL – Target Language

RTT – Round Tip Translation

LIST OF FIGURES

FIGURE 2.1: Bernard Vauquois' Pyramid Showing Comparative Depths of Intermediary Representation.	9
FIGURE 3.1: Diagram of the System's Architecture	25
FIGURE 3.2: Block Diagram of Translation Process.	26
FIGURE 3.3: The Operational Flow Diagram of the Application.	27
FIGURE 3.4: Database of the translator in its notepad form.	28
FIGURE 4.1: System GUI ready to take English words.	31
FIGURE 4.2: System GUI when it takes it's the English word.	32
FIGURE 4.3: Sample of outputs generated by the translation system	33
FIGURE 4.4: Sample of outputs generated by the translation system.	34
FIGURE 4.5: Figure showing what the system database looks like	35

LIST OF TABLES

Table 4.1: Results of system evaluation	36
Table 4.2: Table showing the statistics of Dataset	37

TABLE OF CONTENTS

CERTIFICATION	i
DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
LIST OF ACRONYMS	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
TABLE OF CONTENTS	ix

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND STUDY.	1
1.2 PROBLEM STATEMENT.	2
1.3 AIM AND OBJECTIVES	3
1.4 SCOPE OF STUDY.	3
1.5 PROJECT METHODOLOGY	3
1.6 SIGNIFICANCE OF STUDY	4

CHAPTER TWO

LITERATURE REVIEW

2.1 HISTORY OF MACHINE TRANSLATION	6
------------------------------------	---

2.2 LEVELS OF MACHINE TRANSLATION.	7
2.2.1 METAPHRASE	7
2.2.2 PARAPHRASE	8
2.3 METHODS OF MACHINE TRANSLATION.	8
2.3.1 RULE BASED MACHINE TRANSLATION	8
2.3.1.1 Direct/ dictionary based machine translation	9
2.3.1.2 Transfer based machine translation.	10
2.3.1.3 Interlingua based translation.	10
2.3.2 STATISTICAL MACHINE TRANSLATION.	10
2.3.2.1 Problems associated with statistical machine translation	11
2.3.2.3 Disadvantages of statistical machine translation approach	12
2.3.3. HYBRID BASED TRANSLATION	12
2.3.4 EXAMPLE BASED MACHINE TRANSLATOR [EBMT]	13
2.4 STRUCTURE OF YORUBA AND ENGLISH LANGUAGE	13
2.5 EVALUATION OF MACHINE TRANSLATION SYSTEMS.	15
2.5.1 Round-trip translation (RTT)	15
2.5.2 Human evaluation	16
2.5.3. Automatic evaluation	17
2.6 RELATED WORKS	20

CHAPTER THREE

METHODOLOGY

3.1 APPROACH	24
3.2 REQUIREMENT ANALYSIS	24
3.3 DEVELOPMENT TOOLS	24
3.4 SYSTEM'S ARCHITECTURE	25
3.5 PROCESSES THAT WAS INVOLVED IN THIS PROJECT.	26
3.6 SYSTEM DESIGN ANALYSIS	26
3.7 DATABASE DESIGN	28
3.8 SYSTEM SOFTWARE DESIGN AND IMPLEMENTATION	29

CHAPTER FOUR

SYSTEM EVALUATION, RESULTS AND DISCUSSION

4.1 EVALUATION OF SYSTEM	30
4.2 SYSTEM OUTPUT RESULTS FOR SYSTEM	30
4.3 RESULTS DISCUSSION	36

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION	38
5.2 RECOMMENDATION	38

REFERENCES.

39

Appendix

42

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND STUDY

Language is the basic medium of communication. Language is any formal system of gestures, signs, sounds, and symbols used or conceived as a means of communicating thoughts. Language is ultimately important because it is the primary means through which humans have the ability to communicate and interact with one another (Mayell, 2003). The means by which information is shared across various languages is called TRANSLATION.

Translation has been made very important so as to bridge the gap of information inequality. A lot of countries and organizations rely on translations to disseminate information. The limitation of human translators such as speed, insecurity and high cost has led to the development of machine translators (Oladosu, *et al.*, 2016).

Machine translation (MT) can be defined as a subfield of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another (Arnold, Balkan, Meijer, Humphreys, & Sadler, 1994). It uses computers to automate some or all of the process of translating from one language to another. Machine Translation (MT) has been a focus of investigations by linguists, psychologists, philosophers, computer scientists and even engineers. It has come to be understood as an economic necessity, considering that the growth of international communication keeps intensifying both at government (European Union EU, United Nations Organization UN) and business and commerce levels (exporters need product documentation in the languages of the countries where their products are marketed) (Folajimi & Omonayin, 2012).

Statistical machine translators (SMTs) were first introduced by IBM in 1990, this translators worked by analyzing similar texts in two languages and tries to understand their patterns rather

than using rules and linguistics (Prestov, 2018). The first statistical translation systems worked by splitting the sentence into words, since this approach was straightforward and logical. IBM's first statistical translation model was called Model one and over time the machine has evolved and they can now come as phrase-based models.

1.2 PROBLEM STATEMENT

The growth of international communications has increased to a reasonable level at government, business and commerce levels. Therefore the demand for faster and cheaper translations is in high demand. Traditionally, translation was carried out by human translators but there are certain limitations associated with it which include: high cost, lower speed of translation, undependability and insecurity of confidential information as well as lack of in depth understanding of a language (Oladosu, *et al.*, 2016).

Yoruba language which is one of the mostly spoken languages in Nigeria with over 25 million speakers in the south-western part of the country is endangered and the culture is gradually going into extinction, this call for an immediate need for modern day tools to help the language catch up with technological growth.

In light of all these, an English to Yoruba statistical machine translator was developed to overcome the shortcomings of human translator and help Yoruba language catch up with technological growth.

1.3 AIM AND OBJECTIVES

The aim of this project is to “develop an English to Yoruba machine translation system using direct based approach.”

The objectives of this project are to:

1. Design of a machine translator system using direct approach.
2. Implement the designed system for English to Yoruba translation.
3. Evaluate the effectiveness of the designed system

1.4 SCOPE OF STUDY

This project is focused on creating a machine translation system that translates English words to Yoruba. Direct based approach for machine translation would be used to develop this system and the data for this work would be obtained from locally spoken words and dictionaries. The performance of this system would be evaluated using human evaluation and judgment.

1.5 PROJECT METHODOLOGY

The methods to be followed in creating the translator include:

- **Data creation:** this creates the data that is required to work with the system and it is extracted from locally spoken words.
- **System training:** The corpus to be used is trained to understand the rules of the translation especially grammatically.
- **Programming the translator:** Python is the core language used in developing the translator system.
- **Designing the application's GUI:** the system GUI is the link between the user and the translator system.

1.6 SIGNIFICANCE OF STUDY

The significance of machine translation cannot be overemphasized in this rapidly growing globalization. It can translate contents quickly and provide quality outputs while saving humans the stress, time and cost of human translator and pouring over translation books. Also, MTs are highly confidential and it makes it more favorable. It is accessible everywhere unlike human translators who might not be opportune to go everywhere with you (Folajimi & Omonayin, 2012). The system possesses remarkable ability to translate the content quickly and provides quality outputs, thus saving human the stress and time of poring on translating books or looking for human translator.

In any translation, whether human or automated, the meaning of a text in the source language must be fully transferred to its equivalent meaning in the target language's translation. While on the surface this seems straightforward, it is often far more complex.

It is way more expensive and complex for human translators as no two individual translators will produce identical translations of the same text in the same language pair (Prestov, 2018), and it may take several rounds of revisions to meet the client's requirements.

Translation is never a mere word-for-word substitution.

A human translator must interpret and analyze all of the elements within the text and understand how each word may influence the context of the text. This requires extensive expertise in grammar, syntax (sentence structure), semantics (meanings), etc., in the source and target languages, as well as expertise in the domain.

However, machine translation is comparatively cheap. Confidentiality is another matter which makes machine translation favorable. Giving sensitive data to a human translator might be risky while with machine translation your information is protected. Machine translation has been integrated in many applications. The areas of application include:

Electronic-learning: for learning various languages and tutoring for curricular activities.

Electronic-health: MT has been used widely in hospitals to break communication barriers between health care givers and patients.

Production of technical documents: MT systems can be used for production of technical documents.

Localization of software: MT can be applied in software localization by making available supporting documentation for new software in many languages.

Other applications include: speech translation, information retrieval and information extraction.

It can also be applied in government organizations for the translation of internal documents and assisting administrators in composing texts in non-native language

CHAPTER TWO

LITERATURE REVIEW

2.1. HISTORY OF MACHINE TRANSLATION

Machine translation was first dreamt of in the seventeenth century but it did not come to reality until late twentieth century (Hutchins, 1995). The history of machine translation can be traced from the early systems of the 1950s and 1960s, the impact of the ALPAC report in the mid-1960s, the revival in the 1970s, the appearance of commercial and operational systems in the 1980s, research during the 1980s, new developments in research in the 1990s, and the growing use of systems in the past decade.

The use of MT accelerated in the 1990s. The increase has been most marked in commercial agencies, government services and multinational companies, where translations are produced on a large scale, primarily of technical documentation.

Machine Translation (MT) of natural human languages is not a subject about which most scholars feel neutral. The field has had a long, colorful career. During its first decade in the 1950's, interest and support was fueled by visions of high-speed high-quality translation of arbitrary texts. During its second decade in the 1960's, disillusionment crept in as the number and difficulty of the linguistic problems became increasingly obvious, and as it was realized that the translation problem was not as amenable to automated solution as had been thought.

The climax came with the delivery of the National Academy of Sciences ALPAC report in 1966, condemning the field and, indirectly, its workers alike. The ALPAC report was criticized as narrow, biased, and short-sighted, but its recommendations were adopted and as a result MT projects were cancelled in the U.S. and elsewhere around the world. By 1973, the early part of the third decade of MT, only three government-funded projects were left in the U.S., and by late 1975 there were none. Paradoxically, MT systems were still being used by various

government agencies here and abroad, because there was simply no alternative means of gathering information from foreign [Russian] sources so quickly; in addition, private companies were developing and selling MT systems based on the mid-60's technology so roundly castigated by ALPAC. Nevertheless the general disrepute of MT resulted in a remarkably quiet third decade (Slocum, 1985).

We are now into the fourth decade of MT, and there is a resurgence of interest throughout the world plus a growing number of MAT (Machine-aided Translation) systems in use by governments, business and industry. Industrial firms are also beginning to fund MAT projects of their own, thus it can no longer be said that only government funding keeps the field alive.

The realization that MTs can be useful though imperfect and its capabilities lie beyond what was possible one decade ago.

2.2 LEVELS OF MACHINE TRANSLATION

Machine translation is one of the research areas under computational linguistics, various methods have been created to automate its process and in general the process of translation has two levels; the meta-phrase and the Para-phrase.

2.2.1 METAPHRASE

It means translating word to word. Every translated word is a literal translation of the word. Although the translated text might differ from the meaning of the original text, it means that sometimes the semantics may differ.

2.2.2 PARAPHRASE

This is not a word to word translation unlike the Meta phrase. The translated text would still contain the meaning of the original text.

2.3 METHODS OF MACHINE TRANSLATION

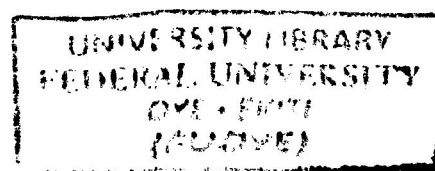
MTs are classified into seven broad categories: rule based, statistical based, hybrid based, example based, knowledge based, principle based and online interactive methods. Most MT researches these days are based on statistical and example based.

2.3.1 RULE BASED MACHINE TRANSLATION

Rule Based Machine Translation (RBMT) has much to do with the morphological, syntactic and semantic information about the source and target language. RBMT is very maintainable and has the ability to deal extensible with the needs of linguistics phenomena. Exceptions in grammar can add difficulties to the system. Its main objective is to convert from source language to target language structures.

The RBMT method uses three approaches including: transfer based MT, dictionary based MT and Interlingua MT.

The disadvantages or problems associated with RBMTs includes; adapting to new domains are usually costly and most times, does not pay off, dealing with rule interactions like ambiguity or idioms in big systems is hard, some information still needs to be set manually. The process is not fully automated and building new dictionaries are really expensive; hence, there are no sufficient dictionaries available (Okpor, 2014).



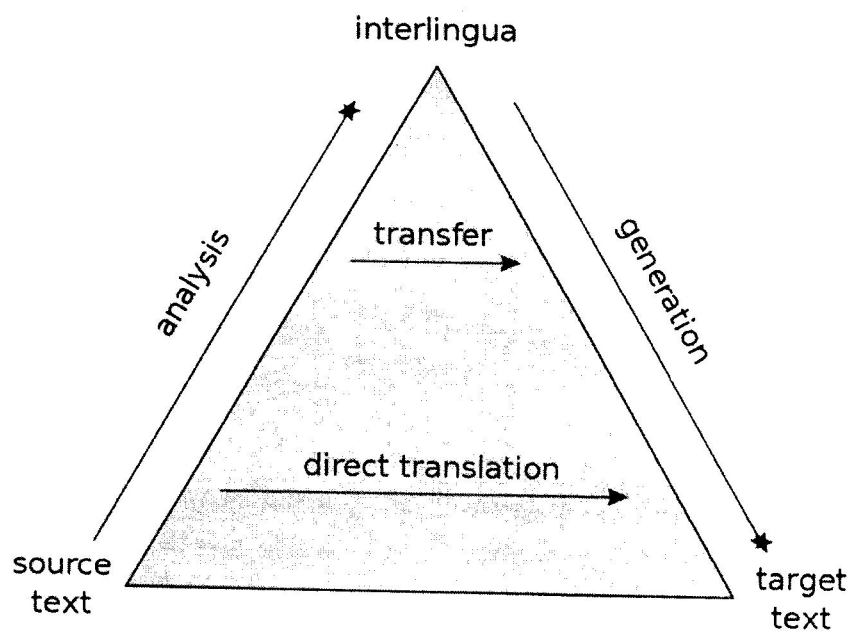


FIGURE 2.1: Bernard Vauquois's pyramid showing comparative depths of intermediary representation. (Source: http://en.wikipedia.org/wiki/Machine_translation#Interlingual).

2.3.1.1 Direct/ Dictionary based machine translation

The first generation of machine translations (late 1940s to mid-1960s) was based on machine readable or electronic dictionaries (Tripathi & Sarkhel, 2010). This method of translation is based on language dictionaries; this method is good in translating words and to some extent phrases but it is not helpful in translating sentences.

Machine translation systems that use this approach are capable of translating a language, called source language (SL) directly to another language, called target language (TL) without passing through an additional/intermediary representation (Bidjmol and John, 2015).

2.3.1.2 TRANSFER BASED MACHINE TRANSLATION

Transfer model belongs to the second generation of machine translation (mid 60s to 1980s). Transfer based systems converts text into a less language specific representation. Structurally, transfer systems can be broken down into three different stages; Analysis, Transfer and Generation. In the first stage, Analysis of the source text is done based on linguistic information such as morphology, part-of-speech, syntax, semantics, etc.

In the second stage, the syntactic/semantic structure of source language is then transferred into the syntactic/semantic structure of the target language (Alejandro & Beatriz, 2013). In the final stage, this module replaces the text in the source language to the target language equivalents.

2.3.1.3 INTERLINGUA BASED TRANSLATION

This model is indented to make linguistic homogeneity across the world. In this method, source language is translated into an intermediary representation which does not depend on any language. The main property of this model is single representation for different languages and much easier to multilingual machine translation. Target language is derived from this auxiliary form of representation (Prasad & Muthukumaran, 2013).

2.3.2 STATISTICAL MACHINE TRANSLATION

The statistical translation methods include:

STATISTICAL WORD-BASED TRANSLATION MODEL

Its fundamental unit is word, its reordering is done using algorithms related to alignment of words are required to achieve utmost accuracy in sentence translation and compound words are idioms, homonyms create complexity for simple word based translation

STATISTICAL PHRASE-BASED MODEL

Its fundamental unit is a phrase or sequence of words. A sequence of words in the source and the target language is developed. Decoding is done based on the vector of features with matching values for the language sequence pair (Zens, Och, & Ney, 2002).

STATISTICAL SYNTAX-BASED MODEL

Its Fundamental unit is the translation rule. **The** translation rule consists of sequence of words and variables in the source language, a syntax tree in the target language (having words or variables at leaves), and a vector of feature values which describes the language pair's likelihood (DeNeefe, Knight, Wang, & Marcu, 2010).

2.3.2.1 PROBLEMS ASSOCIATED WITH STATISTICAL MACHINE

TRANSLATION

1. Sentence Alignment

In parallel corpora single sentences in one language can be found translated into several sentences in the other and vice versa.

2. Statistical Anomalies

Real-world training sets may override translations of, say, proper nouns. An example would be that "I took the bus to Ekiti" gets mis-translated as "I took the cab to Ekiti" due to an abundance of "cab to Ekiti" in the training set.

3. Data Dilution

This is a common anomaly caused when attempting to construct a new statistical model (engine) to represent a distinct terminology (for a specific corporate brand or domain). Training sets used from alternative sources to the specific brand to compensate for a limited quantity of brand-specific corpora may dilute brand terminology, choice of words, text format and style.

4. Idioms

Depending on the corpora used, idioms may not translate "idiomatically".

5. Different word orders:

Order of words in languages might differ. Some classification can be done by naming the typical order of subject (S), verb (V) and object (O) in a sentence and one can talk, for instance, of SVO or VSO languages. There are also additional differences in word orders, for instance, where modifiers for nouns are located, or where the same words are used as a question or a statement (Okpor, 2014).

2.3.2.3 DISADVANTAGES OF STATISTICAL MACHINE TRANSLATION

APPROACH

The disadvantages of statistical machine translation approach includes; corpus creation can be costly for users with limited resources, the results are unexpected. Superficial fluency can be deceiving and statistical machine translation does not work well between languages that have significantly different word orders (e.g. Japanese and European languages).

2.3.3. HYBRID BASED TRANSLATION

HMT takes the advantages of RBMT and Statistical Machine Translation. Hybrid approach to machine translation is an improvement on single approach because it combines the rule based approach with statistical approach. It uses RBMT as baseline and refines the rules through statistical models. Rules are used to pre-process data in an attempt to better guide the statistical engine (Prasad & Muthukumaran, 2013).

Hybrid model differ in various ways:

(a) Rules post-processed by Statistics

Rule based tool is used for translation at first. Statistical model is applied to adjust the translated output of rule based tool. (Prasad & Muthukumaran, 2013)

(b) Statistics guided by rules

In this method, rules are applied to pre-process input that gives better guidance to statistical tool. Rules are also used to post-process the statistical output that caused to normalized output. This method has more flexibility, power and control at the translation time (Prasad & Muthukumaran, 2013)

2.3.4 EXAMPLE BASED MACHINE TRANSLATOR [EBMT]

This method is also called as Memory based translation in which set of sentences from source language is given and generates corresponding translations of target language with point to point mapping was first proposed by Makoto Nagao in 1981. Here examples are used to convert similar types of sentences and previously translated sentence repeated, the same translation is likely to be correct again.

Advantages of an EBMT system over an SMT system (Frederking, 2007)

Its advantages includes; it can work with small set of data (even with one sentence pair), trains translation program and decodes more quickly, less principled (at least in theory).

2.4 STRUCTURE OF YORUBA AND ENGLISH LANGUAGE.

According to (Alejandro & Beatriz, 2013), the Yoruba language structure is as follows.

Syllable structure: Yoruba syllables are open i.e. they all end in a vowel. The most frequent are those formed by a single vowel or by consonant plus vowel. Each syllable has a distinctive tone. Consonant clusters are not permitted but long vowels are possible.

Vowels: The vowel system has seven oral and four nasalized vowels (marked with a tilde) though the nasal vowel [ẽ] occurs very infrequently. Nasalization is phonemic.

Vowel harmony: Yoruba has a limited vowel harmony: a high-mid vowel [e, o] cannot coexist with a low-mid vowel [ɛ, ɔ] in the same word.

Consonants (17-19): The consonantal system is distinguished by its lack of p-sound and by the occurrence of two doubly articulated stops called labio-velars. Labio-velar consonants are very rare outside Africa but are found in some Nilo-Saharan languages and non-Bantu members of Niger-Congo. Besides, Yoruba has a syllabic nasal whose pronunciation depends on the following sound; if it is a vowel, the syllabic nasal is pronounced as a velar ŋ. When the following element is a consonant, the syllabic nasal is articulated at the same place.

Tones: Yoruba has three tones: high (marked by an acute accent), mid (unmarked or marked with a macron), and low (marked by a grave accent). The high tone cannot occur on a word initial vowel. When high tone follows a low tone, the high tone is realized as a rising tone. When a low tone follows a high tone the low tone is realized as a falling tone. Tones serve to distinguish lexical items and, sometimes, grammatical features.

Stress: is evenly distributed.

Script and Orthography: Yoruba is written in a form of the Latin alphabet comprised of 25 letters (below each of them its equivalent in the International Phonetic Alphabet is shown):

A a B b D d E e Ė ė F f G g GB gb H h I i J j K k L l
 [a] [b] [d] [e] [ė] [f] [g] [gb̄] [h] [i] [dʒ] [k] [l]

M m N n O o Ȯ ȯ P p R r S s Ṣ ṣ T t U u W w Y y
 [m] [n̄] [o] [ȯ] [kp] [r] [s] [ʃ] [t] [u] [w] [j]

1. The mid-low vowels ε and ɔ are represented, respectively, by ė and ȯ.
2. When a nasalized vowel immediately follows an oral consonant it is represented by adding n after the vowel but if the nasalized vowel follows a nasal consonant it is not otherwise marked.

3. The labio-velar \widehat{kp} is written p. The letter p is, otherwise, not required because the [p] sound doesn't exist in Yoruba.
4. The labio-velar \widehat{gb} is written gb.
5. The affricate $dʒ$ is represented by j.
6. The fricative f is represented by s.

Lexicon: Yoruba has borrowed many words from Hausa (some originally Arabic), Igbo and English. Like many other African languages, it has ideophones which are a special class of words with particular sound characteristics associated with vivid sensory or mental experiences. In Yoruba, ideophones are made by reduplication and consist of up to four repeated units. Relatives, old people and people in authority should be addressed with politeness which dictates the choice of pronouns, names and nicknames.

Basic Vocabulary: There are two sets of 1-10 numerals: a basic set (listed first) used for counting and a full set used as nouns or adjectives (listed second). The numerals of the basic set have low-tone initials; in the full set an initial m is added and the low-tone changes to a high-tone (except for 1 which drops the first vowel).

2.5 EVALUATION OF MACHINE TRANSLATION SYSTEMS

2.5.1 Round-trip translation (RTT)

It is also known as back-and-forth translation or bi-directional translation, is the process of translating a word, phrase or text into another language (forward translation), then translating the result back into the original language (back translation), using a machine translation (MT) software. It is often used by laypeople to evaluate a machine translation system (Zaanen, Menno, Zwarts, & Simon, 2006) or to test whether a text is suitable for MT (Gaspari & Federico, 2006) when they are unfamiliar with the target language.

One of the problems with this technique is that if there is a problem with the resulting text it is impossible to know whether the error occurred in the forward translation, in the back translation, or in both. In addition it is possible to get a good back translation from a bad forward translation (Somers & Harold, 2005)

2.5.2 Human evaluation

It covers two of the large scale evaluation studies that have had significant impact on the field and they are: the ALPAC study and the ARPA study (White J. O., 1994).

1. Automatic Language Processing Advisory Committee (ALPAC):

One of the constituent parts of the ALPAC report was a study comparing different levels of human translation with machine translation output, using human subjects as judges. The human judges were specially trained for the purpose. The variables studied were "intelligibility" and "fidelity". Intelligibility was a measure of how "understandable" the sentence was, and was measured on a scale of 1–9. Fidelity was a measure of how much information the translated sentence retained compared to the original, and was measured on a scale of 0–9. The study concluded that, "highly reliable assessments can be made of the quality of human and machine translations" (ALPAC, 1966).

2. Advanced Research Projects Agency (ARPA):

As part of the Human Language Technologies Program, the Advanced Research Projects Agency (ARPA) created a methodology to evaluate machine translation systems, and continues to perform evaluations based on this methodology. The evaluation programme involved testing several systems based on different theoretical approaches; statistical, rule-based and human-assisted. A number of methods for the evaluation of the output from these systems were tested in 1992 and the most recent suitable methods were selected for inclusion in the programmes

for subsequent years. The methods were; comprehension evaluation, quality panel evaluation, and evaluation based on adequacy and fluency.

Comprehension evaluation aimed to directly compare systems based on the results from multiple choice comprehension tests, as in Church et al. (1993). The texts chosen were a set of articles in English on the subject of financial news. These articles were translated by professional translators into a series of language pairs, and then translated back into English using the machine translation systems. It was decided that this was not adequate for a standalone method of comparing systems and as such abandoned due to issues with the modification of meaning in the process of translating from English.

Measuring systems based on adequacy and fluency, along with informativeness is now the standard methodology for the ARPA evaluation program (White J. , 1995)

2.5.3. Automatic evaluation

In automatic evaluation, a metric is a measurement. A metric that evaluates machine translation output represents the quality of the output. The quality of a translation is inherently subjective, there is no objective or quantifiable "good." Therefore, any metric must assign quality scores so they correlate with human judgment of quality. That is, a metric should score highly translations that humans score highly, and give low scores to those humans give low scores. Human judgment is the benchmark for assessing automatic metrics, as humans are the end-users of any translation output. The measure of evaluation for metrics is correlation with human judgment. This is generally done at two levels, at the sentence level, where scores are calculated by the metric for a set of translated sentences, and then correlated against human judgment for the same sentences. And at the corpus level, where scores over the sentences are aggregated for both human judgments and metric judgments, and these aggregate scores are then correlated. Figures for correlation at the sentence level are rarely reported, although Banerjee

et al. (2005) do give correlation figures that show that, at least for their metric, sentence level correlation is substantially worse than corpus level correlation.

The aim of this subsection is to give an overview of the state of the art in automatic metrics for evaluating machine translation.

1. BLEU:

It was one of the first metrics to report high correlation with human judgments of quality. The metric is currently one of the most popular in the field. The central idea behind the metric is that "the closer a machine translation is to a professional human translation, the better it is" (Papineni, Roukos, Ward, & Zhu, 2002). The metric calculates scores for individual segments, generally sentence then averages these scores over the whole corpus for a final score. It has been shown to correlate highly with human judgments of quality at the corpus level

BLEU uses a modified form of precision to compare a candidate translation against multiple reference translations. The metric modifies simple precision since machine translation systems have been known to generate more words than appear in a reference text. No other machine translation metric is yet to significantly outperform BLEU with respect to correlation with human judgment across language pairs. (Graham & Baldwin., 2014)

2. The NIST metric:

It is based on the BLEU metric, but with some alterations. Where BLEU simply calculates n-gram precision adding equal weight to each one, NIST also calculates how informative a particular n-gram is. That is to say when a correct n-gram is found, the rarer that n-gram is, the more weight it is given (Dodington, 2012). NIST also differs from BLEU in its calculation of the brevity penalty, insofar as small variations in translation length do not impact the overall score as much.

3. The Word error rate (WER):

It is a metric based on the Levenshtein distance, where the Levenshtein distance works at the character level, WER works at the word level. It was originally used for measuring the performance of speech recognition systems, but is also used in the evaluation of machine translation. The metric is based on the calculation of the number of words that differ between a piece of machine translated text and a reference translation.

A related metric is the Position-independent word error rate (PER), this allows for re-ordering of words and sequences of words between a translated text and a references translation.

4. The METEOR metric:

It is designed to address some of the deficiencies inherent in the BLEU metric. The metric is based on the weighted harmonic mean of unigram precision and unigram recall. The metric was designed after research by Lavie (2004) into the significance of recall in evaluation metrics. Their research showed that metrics based on recall consistently achieved higher correlation than those based on precision alone. METEOR also includes some other features not found in other metrics, such as synonymy matching, where instead of matching only on the exact word form, the metric also matches on synonyms.

5. LEPOR:

A new MT evaluation metric LEPOR was proposed as the combination of many evaluation factors including existing ones (precision, recall) and modified ones (sentence-length penalty and n-gram based word order penalty). The experiments were tested on eight language pairs from ACL-WMT2011 including English-to-other (Spanish, French, German and Czech) and the inverse, and showed that LEPOR yielded higher system-level correlation with human judgments than several existing metrics such as BLEU, Meteor-1.3, TER, AMBER and MP4IBM1 (Han, Wong, & Chao, 2012).

2.6 RELATED WORK

Research has revealed that sentences do not translate languages efficiently, thereby requiring a more refined approach. As a matter of fact, the first successful SMT systems worked on word level translation (Brown, Della, Della, & Mercer, 1990). However, some important developments have evolved and have found their way into machine translation. Example of these developments that relates to this work includes:

In the works of Papineni, Roukos, Ward, & Zhu (2002), BLEU (Bilingual Evaluation Understudy) was described as an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is"

Koehn, Och, & and Marcu (2003), describes Moses as an open source toolkit that contains all required mechanisms for developing a phrase-based SMT system, and merely depends only relying on peripheral tools for implementing the language model and word alignment. Some key tools relevant in our SMT are described below.

Eludiora S. (2013), proposed translation processes for translating English to Yoruba. The proposed machine translator can be used to translate only simple sentences. Context-free grammar and phrase structure grammar were used. It uses rule-based approach, the re-write rules were designed for the translation of the Source language to the target translation.

Och (2003), Venugopaland and Vogel (2005), extensively discussed about MERT. The MERT tool is used for minimum error training. This tool has been extended to randomized initial conditions, permuted the model order to deal with the greedy nature of the algorithm, and tune the dynamic parameter range to increase their potential relative impact. It optimizes decoding performance.

Folajimi and Omonayin (2012) worked on Using Statistical Machine Translation as a language Translation tool for understanding Yoruba. Translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. Existing software tool kits were used.

Oladosu and Olamoyegun (2012) developed a Yoruba-English language translator for doctor-patient mobile chat. They were motivated by the need to improve rural-urban health care by reducing communication barrier between semi-illiterate patients and highly educated medical personnel who are of different ethnic background. Results show that the application has a high degree of novelty and relevance with about 60% and 80% scores respectively

Abiola, et al., (2014), worked on Web based English to Yoruba machine translation. In the research, computational models were formulated using finite state automata, which was used to develop a web-based translation system for Noun-phrases in English language to Yoruba language. Linguists were consulted and there was a detailed study of the syntactic structures of both languages with emphasis on noun-phrases. Rules were formulated for the generation of Noun-phrases from English to Yoruba which were specified using context-free grammar.

Abiola, Adetumbi, Fasiku, & Olatunji (2014), proposed a machine translator for English to Yoruba noun phrases. Rule-based approach was used and the automata theory was used in the analysis of the production rules. The system was able to translate some noun phrases. It was evaluated using The Nigeria daily news and the translator was found to be 90% correct.

Odejobi, Eludiora, Akanbi, Iyanda, & Akinade (2015), proposed a system that can assist in teaching and learning Hausa, Yoruba and Igbo. The model was designed to build a system for the learner of the three major indigenous languages in Nigeria. The study considered body parts, plants and animal names. The Yoruba counting system was also considered in the study. This research is still on-going.

Adenekan, Agbeyangi, & Eludiora (2015), proposed a rule-based approach for English to Yoruba machine translation system. The author revised the three approaches to machine translation process and considered the rule-based approach for the translation process.

Eludiora, Agbeyangi, & Fatusin (2015), while experimenting the concept of Yoruba verbs' tone changing, the authors designed different re-write rules that can address possible different Yoruba verbs that can be low toned, high toned or mid toned. The machine translator was designed, implemented and tested with various sentences.

Abiola, Adetunmbi & Oguntimehin, (2015), this paper considered a hybrid approach to English to Yoruba translation. This paper identified the steps a person would take in developing the proposed system. The research is on-going.

Agbeyangi, Eludiora, & Adenekan (2015), developed a system named "English to Yoruba Machine translation system using rule based approach". They concluded that rule-based approach is a good approach for machine translation system. Their research laid emphasis on the popularity of Yoruba language over the other two languages. The data was collected from home domain vocabularies and the re-write rule was verified using Natural language Toolkits (NLTKs)

Akinwale, Adetunmbi, Obe, & Adesuyi (2015), proposed a web-based English to Yoruba machine translation system. Authors considered a data-driven approach to design the translation process. Context-free grammar was considered for the grammar remodeling. The Yoruba language orthography was not properly discussed in the study.

Oladosu, Esan, Ibrahim, Benjamin, Olatayo, & Bolaji (2016) reviewed the two major approaches (single vs. hybrid) to machine translation and provide critique of existing machine translation systems with their merits and demerits. The research concluded that a single

approach to machine translation fails in achieving the satisfactory performance. On the other hand, a hybrid approach combines the strength of two or more approaches to improve the overall quality and fluency of the translation.

CHAPTER THREE

METHODOLOGY

3.1 APPROACH

This project “development of an English to Yoruba machine translation system using direct based approach.” focused on translating English words to Yoruba words conveniently using python programming language to program the translator. The system database was created manually, it contains over 25,000 translated words and they were gathered from locally spoken words and a few Yoruba and English dictionaries.

3.2 REQUIREMENT ANALYSIS

The requirements and specification for the developed machine translator are:

1. Presenting user friendly graphic user interface (GUI) for the application.
2. Input words in English to be translated to Yoruba.
3. Translate the word inputted to its correct equivalent in Yoruba language
4. Output the translated equivalent.
5. Implement the translation system using python programming language with PyQt designer (GUI module).

3.3 DEVELOPMENT TOOLS

The main tools utilized for this project are:

1. Python programming language: this was used as the core programming language for the application development. Python Programming Language was chosen because it has API for natural language processing.
2. PyQt5: this was used for the design of the application GUI.

3. py2exe: a Python extension which converts Python scripts into executable Windows programs, able to run without requiring a Python installation. This was used to compile the python codes (.py) to an executable file (.exe).

3.4 SYSTEM'S ARCHITECTURE

The diagram for the system's architecture is shown below in figure 3.1

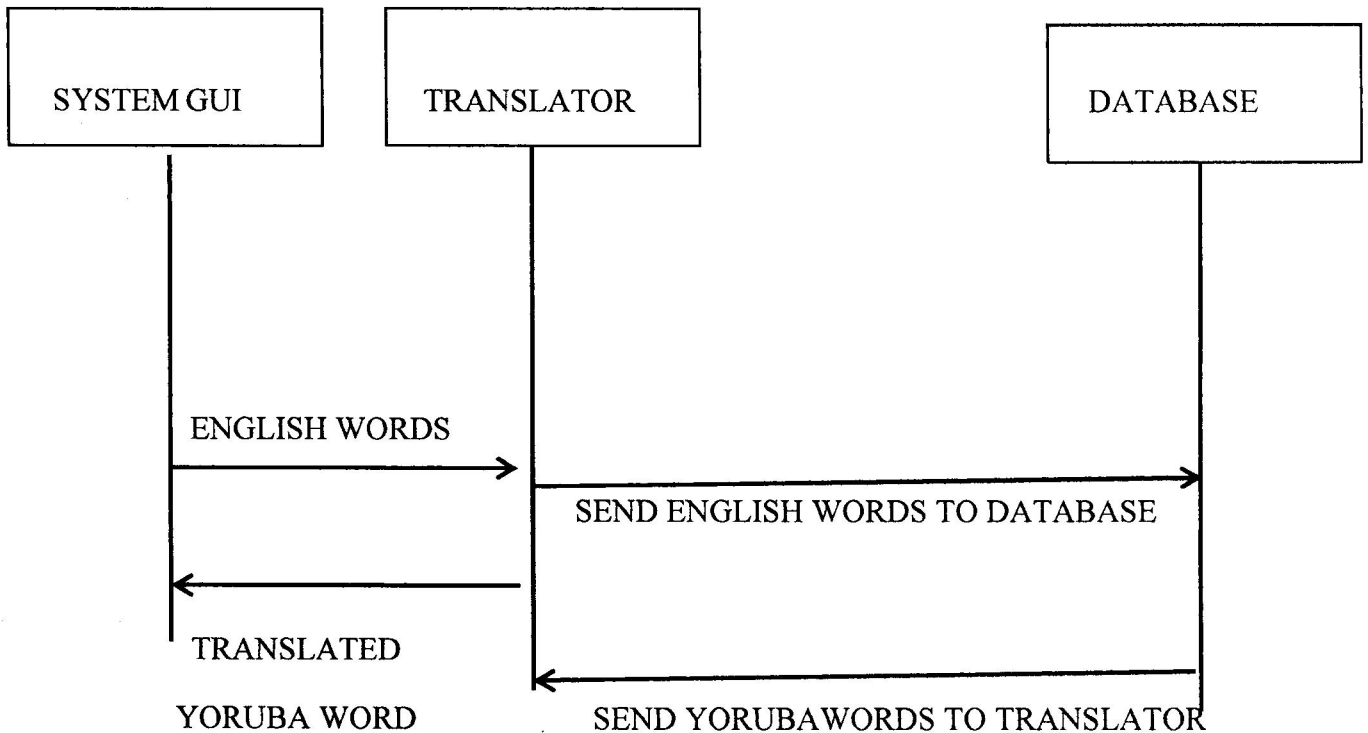


FIGURE 3.1: Diagram of the System's Architecture. [Adapted from: (olaleye, 2017)]

The system architecture of the developed machine translator represented in figure 3.1 above is made up of:

1. The system GUI: this is the user interface that connects the user to the machine translator. It readily allows user to input the English word to be translated and it displays the already translated Yoruba word.
2. The translator: the translator is responsible for correctly mapping the inputted English word provided by the user to its equivalent Yoruba translation fetched from the database.

3. The database: this is the storage for the translation system. It contains over 10,000 translated English words.

3.5 PROCESSES THAT WAS INVOLVED IN CREATING THIS PROJECT

1. Data collection: Here the data that was required to work with the system was created and this data was gotten from locally spoken words and dictionaries.
2. System training: The system was trained to take in words and translate using the direct method of translation. The total number of words used to train the system is 25,000 words.
3. Programming the translator: python programming language was used to implement this model and program the translator as a whole.
4. Designing the system's GUI: the system GUI is the link between the user and the translator system and was designed using PyQt designer.

3.6 SYSTEM DESIGN ANALYSIS

To further illustrate the workings of the system, see figure 3.2 below which explains that when a user inputs a particular English word it goes through a pre-processing stage which involves analyzing the English word to obtain the target language's equivalent word from the database.

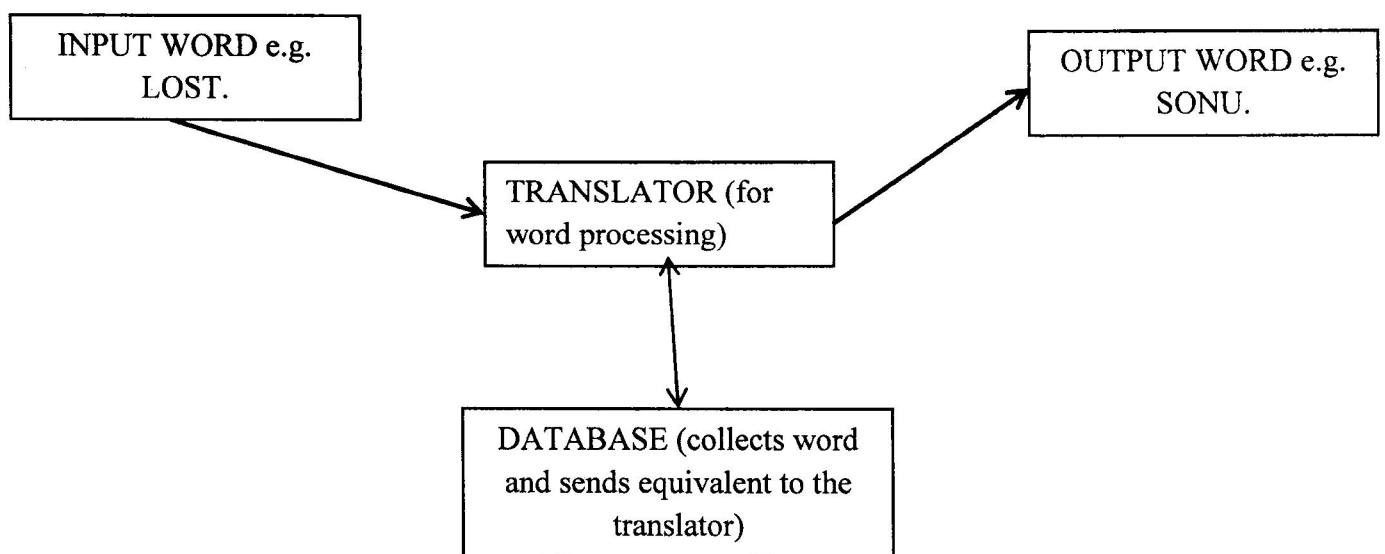


Figure 3.2: Block Diagram of Translation Process.

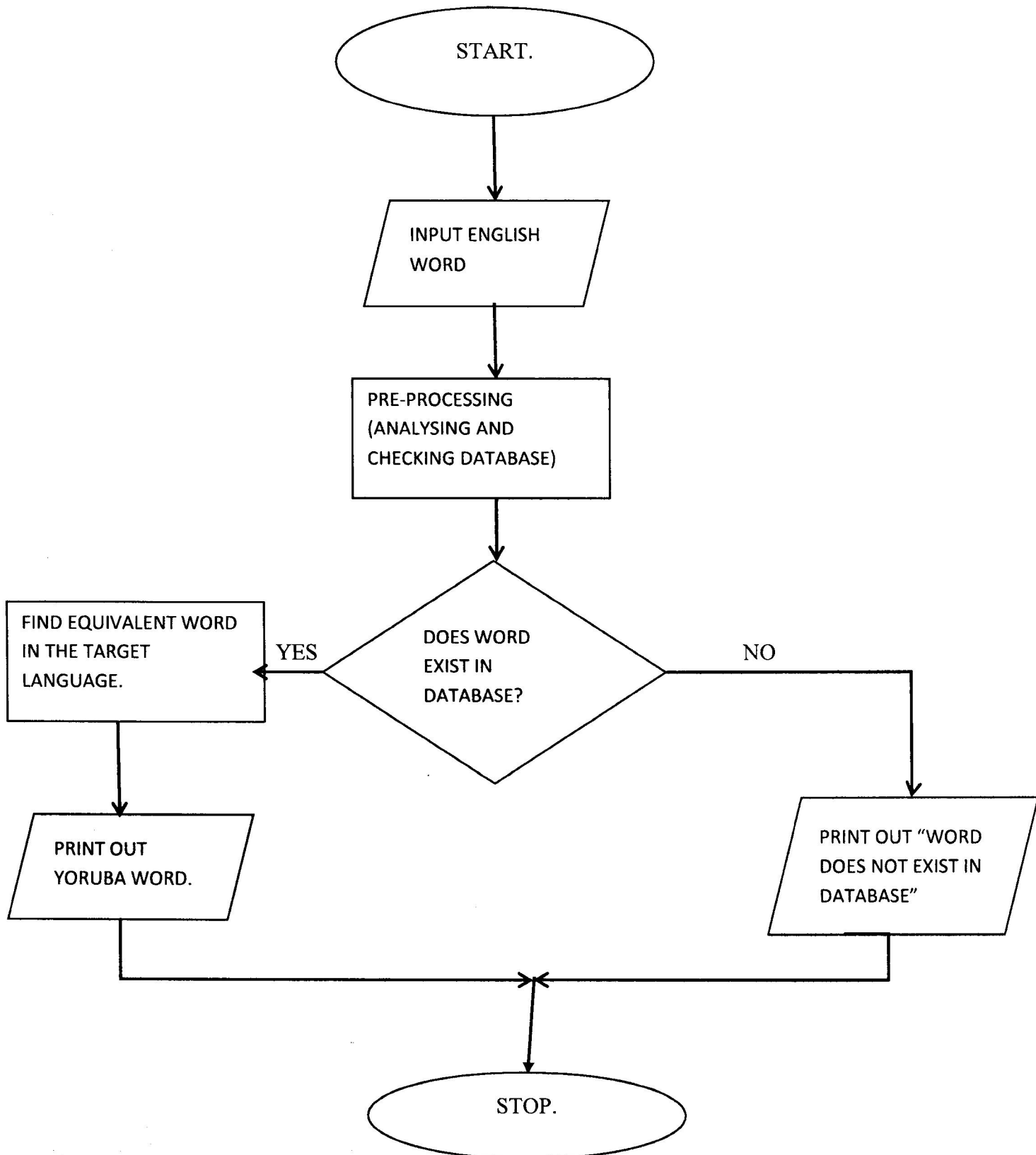


Figure 3.3: flow chart showing how the system works.

3.7 DATABASE DESIGN.

The database was arranged alphabetically with each words figures of speech right beside it and its translation, other translation suggestions is also included.

File	Edit	Format	View	Help
Abandon	v	kò	fi sílẹ̀	kò sílẹ̀
abbreviate	v	gẹ̀	kúrú	
abdomen	n	ikùn		
abduct	v	gbé sálo	fa juro	ré lo
ability	n	agbára	óye	
ablaze	adj	gbiná	jó-wòwò	
able	adj	le	lágbára	lè
abnormal	adj	ṣàṣẹ̀jì	abàmi	
abolish	v	parẹ̀	sọ dasán	sọ dọfo
abortion	n	ṣíṣẹ̀	oyún	ìṣẹ̀yún
abroad	adv	lóde	lèhìn odi	dálẹ̀ kale
abrupt	adj	lójijì	láimòtẹ̀lẹ̀	
abscond	v	salo		
absent	adj	kò sí	kò wá	
absolutely	adj	pátápátá		
absorb	v	fàmu	mì lá	mu tán
abstain	v	fà sèhìn	takété	sẹ̀ ra
abstract	adj	àfòyemò		
abundance	n	òpò		
abuse	v	lò lónà	àitò bú	lò nílòkulò pè lórúk ọkórúkọ
abuse	n	èébú	àbùkù	ilò lónà àitò ilòkulò
academy	n	ilé ẹ̀kò	gíga	ilé ẹ̀kò ijìnlẹ̀
accelerate	v	mú yára	mú lo	síwájú
accent	n	àmì	ohùn	
accept	v	gbà	tẹ̀wògbà	
access	n	àyè	ònà	
accident	n	èyì	àgbákò	jà, bá àdébá
accommodate	v	fùn	láyè	
accompany	v	bá-lo	bá-rìn	bá-kẹ̀gbẹ̀
accomplish	v	ṣ e parí		ṣ e-tán ye à ṣ epé
according	prep	bí	gẹ̀-gẹ̀	bí
account	n	owó àpamò	ìṣìrò	ìkàsí
accountant	n	oníyìrò	akòwé	owó
accumulate	v	kójọ	dá-lé	
accumulation	n	ìkójọ	pò	àdálé àkànmò
accuracy	n	ìṣegẹ̀gẹ̀	ì ṣedédé	
accuse	v	fisùn	kà sí	lòrùn
ache	n	fí-fò	ríro	
achieve	v	ye-tán	ye-parí	
acid	adj	kan	kíkan	mú
acknowledge	v	jẹ̀wò	gbà	
acne	n	eéwo	èrẹ̀kẹ̀	
acquaintance	n	ìfihàn	ojúlùmò	òrẹ̀

Figure 3.4: database of the translator in its notepad form.

3.8 SYSTEM SOFTWARE DESIGN AND IMPLEMENTATION

The software design contains the Graphical User Interface (GUI) which is designed using PyQt5 and implemented using python programming language. The GUI features three phases, the first phase features a textbox (space) that allows you input the English word you are trying to translate, and the second phase is a button that carries the sign “translate” that initiates the translation process. The third phase displays the translated Yoruba word. The fourth phase is a button that displays the database of the system.

On entry of the text in source language (English), the translator module of the code begins to execute. The search for the word commences in the database.

The translator module accepts the inputted word from the GUI module then searches the database module to confirm that the words exist in the database. However, if the words are not in the database an error/ suggestion message will be generated. The translated word in target language (Yoruba) is then displayed by the GUI.

Python programming language was used in the software coding and the interface of the machine is designed using PyQt5. The database was created manually where each word is categorized according to its parts of speech. The machine translation system has the capability to translate words in English language to its equivalent word in Yoruba.

CHAPTER FOUR

SYSTEM EVALUATION, RESULT AND DISCUSSION

4.1 EVALUATION OF SYSTEM

Human judgment approach was used in evaluating this project. The approach involves testing the system with three sets of data gotten from three different sources. Each data set contains about 800 words and the accuracy of the system in each case was recorded. The evaluation was done in other to test the accuracy of the developed system. The three different sources for the dataset used for this evaluation are: 1000 words were extracted randomly from a novel titled "Norwegian woods" by Haruki Murakami (Murakami, 2001), 500 words were extracted randomly from the subtitle file of a movie titled "lion heart" (Nnaji, 2018) and 1000 words were extracted randomly from an anthology titled "The anthology I" (wordslingers, 2017) which tested about 1000 words too.

4.2 SYSTEM RESULTS

The following figures below shows a step by step process from the inputting to the result gotten from the system.

Figure 4.1 shows the translator's GUI ready to take in the English words to be translated, figure 4.2 shows the translator's system with the already inputted English word to be translated, the figure 4.3 and 4.4 shows the translator after translating the English word to its equivalent Yoruba word. Figure 4.5 shows what the database of the system looks like.

Help

ENGLISH - YORUBA WORD TRANSLATOR

Enter text here

Translate

Display

Fig. 4.1 shows the system GUI ready to take English words.

Help

ENGLISH - YORUBA WORD TRANSLATOR

ABANDON
<input type="button" value="Translate"/>
Display

Fig 4.2: System GUI when it takes it's the English word.

The figures 4.3, 4.4 shows the sample of outputs generated by the translation system.

Help

ENGLISH - YORUBA WORD TRANSLATOR

abandon

Translate

Display

Abandon

v

kò

Other Translations

fí sílẹ̀, kọ sílẹ̀

ENGLISH - YORUBA WORD TRANSLATOR

ablaze

Translate

Display

Ablaze

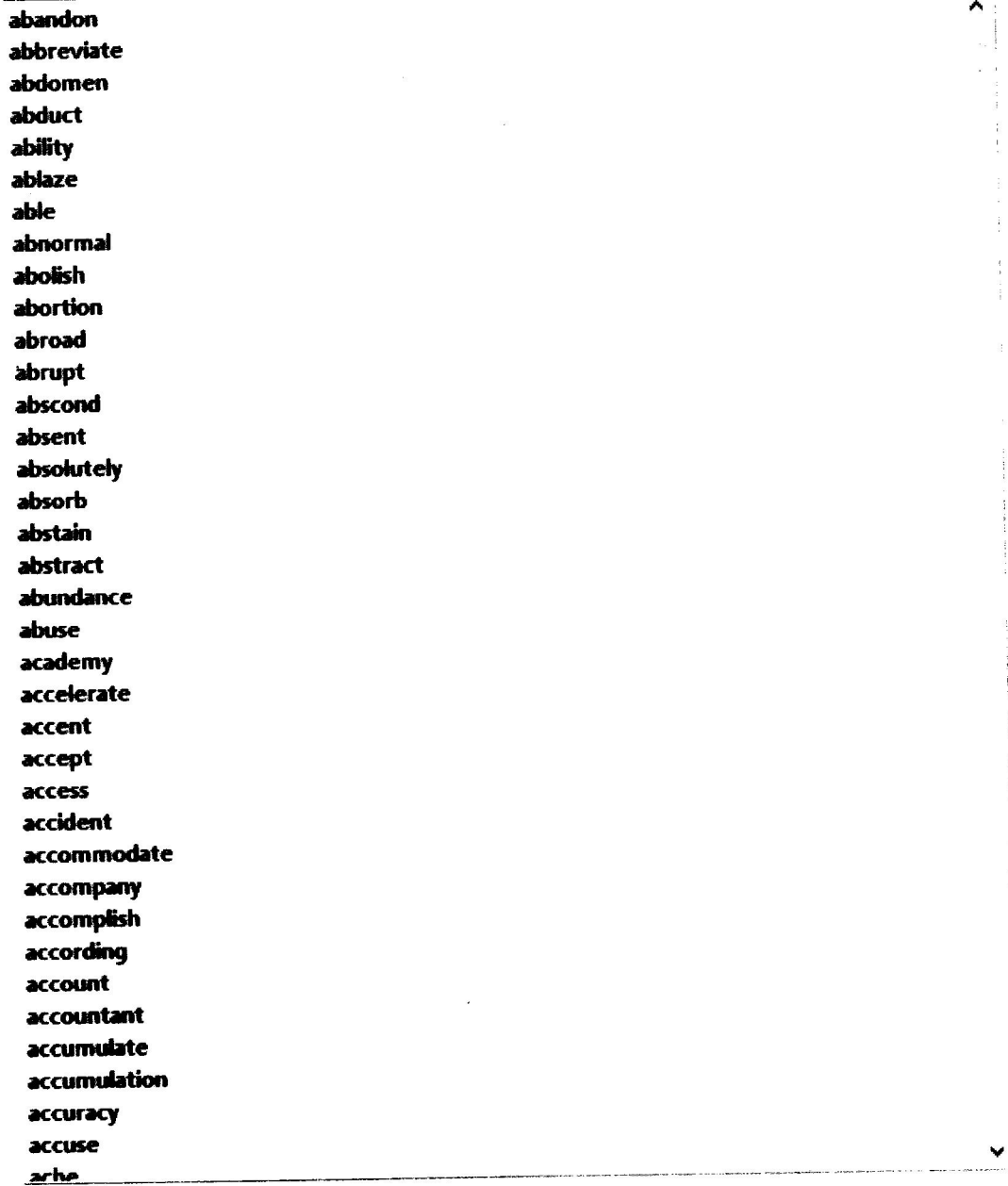
adj

gbiná

Other Translations

jó-wòwò

Word Datase



The image shows a window titled "Word Datase" containing a list of words. The words are listed in a single column, starting with "abandon" at the top and ending with "arhe" at the bottom. The list is enclosed in a rectangular border with a small upward-pointing arrow at the top right and a downward-pointing arrow at the bottom right, suggesting it is a scrollable list.

abandon
abbreviate
abdomen
abduct
ability
ablaze
able
abnormal
abolish
abortion
abroad
abrupt
abscond
absent
absolutely
absorb
abstain
abstract
abundance
abuse
academy
accelerate
accent
accept
access
accident
accommodate
accompany
accomplish
according
account
accountant
accumulate
accumulation
accuracy
accuse
arhe

Figure 4.5: figure showing what the system database looks like.

4.3 DISCUSSION OF RESULT

The system was evaluated to determine the performance of the developed machine translator. The quality and shortcoming of the system designed was verified using its accuracy in translating the words. Most of the data gotten for the system testing were from common sources to ensure that the system was tested with more locally spoken words and common words. The results of the evaluation is shown in Table 4. This is evaluation based on the accuracy of translating the inputted English words, the total number of words tested is 2500, the total number of words found in the system is 2129 and the number of words not found in the system is 246, the accuracy of each dataset is as follows: 92.6%, 87.4%, 89.1%. The accuracy of the developed system is 89.7%

Table 4.1: Results of system evaluation.

	Number of tested words.	Number of words found in system.	Number of words not found.	System accuracy
DATASET 1	1000	926	74	92.6%
DATASET 2	500	437	63	87.4%
DATASET 3	1000	891	109	89.1%
	2500	2129	246	89.7%

Table 2 below shows the results of the developed system, the system was trained with over 25,000 words and tested with 2500 words and the system was found to be 89.7% accurate.

Table 4.2: Table showing the statistics of Dataset.

DATASET	NUMBER
Training	25,000
Test	2500
Accuracy	89.7%

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

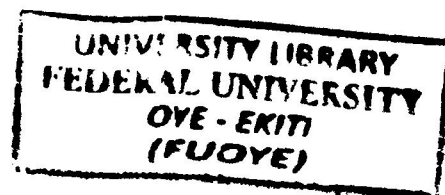
5.1 CONCLUSION

The need for Machine translation system cannot be overemphasized in this era of rapid globalization, especially for indigenous languages in Nigeria, the results gotten show that the people are not good at writing the Yoruba language, this reveals the extent to which Yoruba Language is going into extinction. The English to Yoruba machine translation system was developed. The system was designed to enhance the learning of the Yorùbá language. It is user-friendly and allows learners to learn the language at ease. The evaluation focused on translation accuracy and it was evaluated using 2500 and the accuracy of the system was 89.7%.

5.2 RECOMMENDATION

The English to Yoruba words based translation system was successfully implemented. Accuracy of about 89.7% was achieved. This work only handles the translation of words which is only a minute part of a complete sentence. The system can be developed further to produce acceptable translation of complete sentences and hosted on the internet for public use. The result obtained from this research work reveals that most speakers of Yoruba language have low proficiency at writing it, some even find it hard to differentiate some ambiguous English words from its Yoruba equivalent.

Finally, I would recommend that the government should invest in machine translators to aid socio-economic growth of Yoruba-speaking states.



REFERENCES

- Abiola, O. B., Adetunmbi, A. O., Fasiku, A. I., & Olatunji, K. A. (2014). A Web-Based English to Yoruba noun-phrases Machine Translation System. *International Journal of English and Literature*, 5(3), 71-78.
- Alejandro, G., & Beatriz, A. (2013). <http://www.languagesgulper.com/eng/Yoruba.html>. Retrieved from languagesgulper.
- ALPAC. (1966). *languages and machines: computers in translation and linguistics*. Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Academy of Sciences, National Research Council. . washinton D.C: National Research Council.
- Arnold, D., Balkan, L., Meijer, S., Humphreys, R. L., & Sadler, L. (1994). *Machine Translation: An Introductory Guide* (1st ed.). NCC Blackwell, London: NCC Blackwell Ltd.
- Banerjee, S., & Lavie, A. (2005). *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*.
- Brown, P., Della, V. J., Della, S. A., & Mercer, R. L. (1990). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2), 263-311.
- Church, K., & Hovy, E. (1993). *Good Applications for Crummy Machine Translation*.
- DeNeefe, S., Knight, K., Wang, W., & Marcu, D. (2010). *What can syntax-based MT learn from phrase-based MT?* Retrieved june 22, 2018, from Isi education: <http://www.isi.edu/natural-language/mt/ats-vs-ghkm.pdf>
- Doddington, G. (2012). Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. *Proceedings of the Human Language Technology Conference (HLT)*, (pp. 128–132). San Diego.
- Folajimi, Y. O., & Omonayin, I. (2012). Using Statistical Machine Translation As A language Translation tool for understanding Yoruba. *EIE's 2nd International Conference on Computing Energy, Networking, Robotics and Telecommunications*, (pp. 86-91).
- Frederking, B. (2007). *Example-based MT (EBMT)*. Retrieved june 20, 2018, from CMU: <http://www.cs.cmu.edu/afs/cs/user/alavie/11-731/731-cmt/www/ebmt2007.pdf>
- Gaspari, & Federico. (2006). Look who's translating. Impersonation, Chinese whispers and fun with machine translation on the Internet. *EAMT*, 149-158.
- Graham, Y., & Baldwin., T. (2014). Testing for Significance of Increased Correlation with Human Judgment. qatar.

- Han, A., Wong, D., & Chao, L. (2012). LEPOR: A Robust Evaluation Metric for Machine Translation with Augmented Factors. *Proceedings of the 24th International Conference on Computational Linguistics*, (pp. 441–450). mumbai, india.
- Hutchins, J. (1995). *Example based machine translation - A review and commentary*. Retrieved June 20, 2018, from hutchinweb: <http://www.hutchinsweb.me.uk/MTJ.pdf>
- Koehn, P., Och, F. J., & and Marcu, D. (2003). Statistical Phrase-based Translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (pp. 48-54). Morristown, NJ, USA: Association for Computational Linguistics.
- Lavie, A., Sagae, K., & Jayaraman, S. (2004). The Significance of Recall in Automatic Metrics for MT Evaluation. *Proceedings of AMTA 2004*. Washington DC.
- Mayell. (2003, February 20). *When did "modern" behavior emerge in humans?* Retrieved June 19, 2018, from national geographic: http://news.nationalgeographic.com/news/2003/02/0220_030220_humanorigins2.html
- Moore, A. (2000). <http://www.universalteacher.org.uk/lang/engstruct.htm>. Retrieved from universal teacher.
- Murakami, H. (2001). *norwegian woods*. tokyo: Harvill Press.
- Nnaji, G. (2018, october). *Lionheart. subtitle file*. nigeria.
- Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (pp. 160-167). Morristown, NJ, USA: Association for Computational Linguistics.
- Okpor, M. D. (2014). Machine translation Approaches: Issues and challenges. *International Journal of Computer Science Issue*, 11(2), 159-165.
- Oladosu, J., Esan, A., Ibrahim, A., Benjamin, A., Olatayo, O., & Bolaji, O. (2016). Approaches to Machine Translation: A Review. *FUOYE Journal of Engineering and Technology*, 120-126.
- olaleye, t. (2017). *development of a rule based yoruba to english translator*. federal university oye ekiti, Department of computer engineering . ekiti: FUOYE. Retrieved 2019
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 311-318). Stroudsburg: PA, USA.
- Prasad, T. V., & Muthukumaran, G. (2013). Telugu to English Translation using Direct Machine Translation Approach. *International Journal of Science and Engineering Investigations*, 2(2), 25.

- Prestov, I. (2018, march 19). *history of machine translation from the Cold War to deep learning*. Retrieved june 19, 2018, from Medium: <https://medium.freecodecamp.org/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5>
- Slocum, J. (1985, January-March). A Survey of Machine Translation. *Computational Linguistics*, 11(1), 3.
- Somers, & Harold. (2005). Round-trip translation: What is it good for? *Proceedings of the Australasian Language Technology Workshop ALTW*, (pp. 127–133).
- Tripathi, S., & Sarkhel, J. K. (2010). Approaches to machine translation. *Annals of Library and Information Studies*, 388.
- Venugopal, A., & Vogel, S. (2005). Considerations in mce and mmi training for statistical machine translation. *Tenth Conference of the European Association for Machine Translation*, (pp. 120-123). Budapest, Hungary, May.
- White, J. (1995). *Approaches to Black Box MT Evaluation*.
- White, J. O. (1994). The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Future Approaches. *Proceedings of the 1st Conference of the Association for Machine Translation in the Americas*, (pp. 193–205). columbia.
- wordslingers. (2017). the anthology 1. word slingers.
- Zaanen, Menno, Zwarts, & Simon. (2006). Unsupervised measurement of translation quality using multi-engine, bidirectional translation. *Springer-Verlag*, 1208-1214.
- Zens, R., Och, F., & Ney, H. (2002). Phrase based machine translation. *Lecture Notes in Computer Science*, 35-56.

APPENDIX A.

```
class MyWindow(QtWidgets.QMainWindow):
    def closeEvent(self, event):
        try:
            if QtWidgets.QMessageBox.question(QtWidgets.QMessageBox(), "Exit",
                                             "Are you sure you want to exit?") == QtWidgets.QMessageBox.Yes:
                event.accept()
            else:
                event.ignore()
        except Exception as e:
            self.showMessage("Exception", str(e), "error")
```

```
class Ui_MainWindow(object):
```

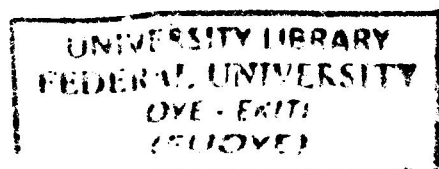
```
    def __init__(self):
        self.matches = []
        try:
            self.dico = smt.loadWords()
        except Exception as e:
            self.showMessage("Error!", "Unable to get Load words.", "warning")
        self.lastword = ""
```

```
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(822, 726)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout_5 = QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout_5.setObjectName("verticalLayout_5")
        self.titleLabel = QtWidgets.QLabel(self.centralwidget)
        font = QtGui.QFont()
        font.setPointSize(20)
        font.setBold(True)
        font.setWeight(75)
        self.titleLabel.setFont(font)
        self.titleLabel.setAlignment(QtCore.Qt.AlignCenter)
        self.titleLabel.setObjectName("titleLabel")
        self.verticalLayout_5.addWidget(self.titleLabel)
        self.rootFrame = QtWidgets.QFrame(self.centralwidget)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
        QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
```

```

sizePolicy.setVerticalStretch(2)
sizePolicy.setHeightForWidth(self.rootFrame.sizePolicy().hasHeightForWidth())
self.rootFrame.setSizePolicy(sizePolicy)
self.rootFrame setFrameShape(QtWidgets.QFrame.StyledPanel)
self.rootFrame setFrameShadow(QtWidgets.QFrame.Raised)
self.rootFrame.setObjectName("rootFrame")
self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.rootFrame)
self.horizontalLayout_2.setObjectName("horizontalLayout_2")
self.mainContainer = QtWidgets.QFrame(self.rootFrame)
self.mainContainer setFrameShape(QtWidgets.QFrame.StyledPanel)
self.mainContainer setFrameShadow(QtWidgets.QFrame.Raised)
self.mainContainer.setObjectName("mainContainer")
self.verticalLayout_3 = QtWidgets.QVBoxLayout(self.mainContainer)
self.verticalLayout_3.setObjectName("verticalLayout_3")
self.input_tf = QtWidgets.QLineEdit(self.mainContainer)
font = QtGui.QFont()
font.setPointSize(12)
self.input_tf.setFont(font)
self.input_tf.setObjectName("input_tf")
self.verticalLayout_3.addWidget(self.input_tf)
self.frame = QtWidgets.QFrame(self.mainContainer)
self.frame setFrameShape(QtWidgets.QFrame.StyledPanel)
self.frame setFrameShadow(QtWidgets.QFrame.Raised)
self.frame.setObjectName("frame")
self.gridLayout = QtWidgets.QGridLayout(self.frame)
self.gridLayout.setObjectName("gridLayout")
self.translateButton = QtWidgets.QPushButton(self.frame)
font = QtGui.QFont()
font.setPointSize(14)
self.translateButton.setFont(font)
self.translateButton.setObjectName("translateButton")
self.gridLayout.addWidget(self.translateButton, 0, 0, 1, 1)
self.verticalLayout_3.addWidget(self.frame)
self.stackedWidget = QtWidgets.QStackedWidget(self.mainContainer)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
QtWidgets.QSizePolicy.Preferred)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(1)
sizePolicy.setHeightForWidth(self.stackedWidget.sizePolicy().hasHeightForWidth())
self.stackedWidget.setSizePolicy(sizePolicy)
self.stackedWidget.setObjectName("stackedWidget")
self.page = QtWidgets.QWidget()
self.page.setObjectName("page")
self.verticalLayout = QtWidgets.QVBoxLayout(self.page)
self.verticalLayout.setObjectName("verticalLayout")

```



```

self.frame2 = QtWidgets.QFrame(self.page)
self.frame2.setFrameShape(QtWidgets.QFrame.StyledPanel)
self.frame2.setFrameShadow(QtWidgets.QFrame.Raised)
self.frame2.setObjectName("frame2")
self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.frame2)
self.verticalLayout_2.setObjectName("verticalLayout_2")
self.suggestion_tf = QtWidgets.QPlainTextEdit(self.frame2)
self.suggestion_tf.setObjectName("suggestion_tf")
font = QtGui.QFont()
font.setBold(True)
font.setPointSize(20)
self.suggestion_tf.setFont(font)
self.verticalLayout_2.addWidget(self.suggestion_tf)
self.buttonFrame = QtWidgets.QFrame(self.frame2)
self.buttonFrame.setFrameShape(QtWidgets.QFrame.StyledPanel)
self.buttonFrame.setFrameShadow(QtWidgets.QFrame.Raised)
self.buttonFrame.setObjectName("buttonFrame")
self.horizontalLayout = QtWidgets.QHBoxLayout(self.buttonFrame)
self.horizontalLayout.setObjectName("horizontalLayout")
self.yesButton = QtWidgets.QPushButton(self.buttonFrame)
font = QtGui.QFont()
font.setPointSize(12)
self.yesButton.setFont(font)
self.yesButton.setObjectName("yesButton")
self.horizontalLayout.addWidget(self.yesButton)
self.noButton = QtWidgets.QPushButton(self.buttonFrame)
font = QtGui.QFont()
font.setPointSize(12)
self.noButton.setFont(font)
self.noButton.setObjectName("noButton")
self.horizontalLayout.addWidget(self.noButton)
self.verticalLayout_2.addWidget(self.buttonFrame)
self.verticalLayout.addWidget(self.frame2)
self.stackedWidget.addWidget(self.page)
self.page_2 = QtWidgets.QWidget()
self.page_2.setObjectName("page_2")
self.verticalLayout_6 = QtWidgets.QVBoxLayout(self.page_2)
self.verticalLayout_6.setObjectName("verticalLayout_6")
self.notification_tf = QtWidgets.QPlainTextEdit(self.page_2)
self.notification_tf.setObjectName("notification_tf")
self.verticalLayout_6.addWidget(self.notification_tf)
self.stackedWidget.addWidget(self.page_2)
self.verticalLayout_3.addWidget(self.stackedWidget)
self.display_groupBox = QtWidgets.QGroupBox(self.mainContainer)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,

```

```

QtWidgets.QSizePolicy.Preferred)
    sizePolicy.setHorizontalStretch(0)
    sizePolicy.setVerticalStretch(1)
    sizePolicy.setHeightForWidth(self.display_groupBox.sizePolicy().hasHeightForWidth())
    self.display_groupBox.setSizePolicy(sizePolicy)
    font = QtGui.QFont()
    font.setPointSize(12)
    self.display_groupBox.setFont(font)
    self.display_groupBox.setObjectName("display_groupBox")
    self.verticalLayout_7 = QtWidgets.QVBoxLayout(self.display_groupBox)
    self.verticalLayout_7.setObjectName("verticalLayout_7")
    self.output_tf = QtWidgets.QTextEdit(self.display_groupBox)
    self.output_tf.setReadOnly(True)
    self.output_tf.setPlaceholderText("")
    self.output_tf.setObjectName("output_tf")
    self.verticalLayout_7.addWidget(self.output_tf)
    self.verticalLayout_3.addWidget(self.display_groupBox)
    self.horizontalLayout_2.addWidget(self.mainContainer)
    self.databaseContainer = QtWidgets.QGroupBox(self.rootFrame)
    font = QtGui.QFont()
    font.setPointSize(12)
    self.databaseContainer.setFont(font)
    self.databaseContainer.setObjectName("databaseContainer")
    self.verticalLayout_4 = QtWidgets.QVBoxLayout(self.databaseContainer)
    self.verticalLayout_4.setObjectName("verticalLayout_4")
    self.dataBaseList = QtWidgets.QListWidget(self.databaseContainer)
    font = QtGui.QFont()
    font.setBold(True)
    font.setWeight(75)
    self.dataBaseList.setFont(font)
    self.dataBaseList.setObjectName("dataBaseList")
    self.verticalLayout_4.addWidget(self.dataBaseList)
    self.horizontalLayout_2.addWidget(self.databaseContainer)
    self.verticalLayout_5.addWidget(self.rootFrame)
    MainWindow.setCentralWidget(self.centralwidget)
    self.menubar = QtWidgets.QMenuBar(MainWindow)
    self.menubar.setGeometry(QtCore.QRect(0, 0, 822, 21))
    self.menubar.setObjectName("menubar")
    self.menuABOUT = QtWidgets.QMenu(self.menubar)
    self.menuABOUT.setObjectName("menuABOUT")
    MainWindow.setMenuBar(self.menubar)
    self.statusbar = QtWidgets.QStatusBar(MainWindow)
    self.statusbar.setObjectName("statusbar")
    MainWindow.setStatusBar(self.statusbar)
    self.actionAbout = QtWidgets.QAction(MainWindow)

```

```

self.actionAbout.setObjectName("actionAbout")
self.actionWord_DataBase = QtWidgets.QAction(MainWindow)
self.actionWord_DataBase.setObjectName("actionWord_DataBase")
self.actionExit = QtWidgets.QAction(MainWindow)
self.actionExit.setObjectName("actionExit")
self.menuABOUT.addAction(self.actionAbout)
self.menuABOUT.addAction(self.actionWord_DataBase)
self.menuABOUT.addAction(self.actionExit)
self.menubar.addAction(self.menuABOUT.menuAction())

```

```

self.retranslateUi(MainWindow)
self.stackedWidget.setCurrentIndex(1)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

```

```

#my own settings
self.translateButton.clicked.connect(self.translate)
self.input_tf.returnPressed.connect(self.translate)
self.yesButton.clicked.connect(self.yesAction)
self.noButton.clicked.connect(self.noAction)
self.actionWord_DataBase.triggered.connect(self.database)
self.loadDataBase()
self.dataBaseList.itemClicked.connect(self.tableAction)
self.actionAbout.triggered.connect(self.about)
self.actionExit.triggered.connect(self.exitAction)

```

```

def exitAction(self):

```

```

    try:

```

```

        if QtWidgets.QMessageBox.question(QtWidgets.QMessageBox(),"Exit","Are you sure you
want to exit?") == QtWidgets.QMessageBox.Yes:

```

```

            sys.exit(0)

```

```

        except Exception as e:

```

```

            self.showMessage("Exception",str(e),"error")

```

```

def about(self):

```

```

    self.showMessage("About","ENGLISH-YORUBA WORD TRANSLATION APPLICATION\nThis
program was developed by ADEBAYO SOORE PROMISE\nMATRIC NO: CPE/13/1071","information")

```

```

def tableAction(self):

```

```

    word = self.dataBaseList.currentItem().text()

```

```

    trnslatn = self.stripList(self.dico[word][1])

```

```

    if len(trnslatn) ==1:

```

```

        self.output_tf.setText("<html>"

```

```

            "<head>"

```

```

            "</head>"

```



```

        "<body>"
        "<h1>%s</h1>"
        "<h3>%s</h3>"
        "<h1>%s</h1>"
        "</body>"
        "</html>"%(word.title(),self.dico[word][0], trnslatn[0]))
    #self.output_tf.setPlainText(word.title() + " " + self.dico[word][0] + "\n" +
self.stripList(self.dico[word][1]))
    else:
        self.output_tf.setText("<html>"
            "<head>"
            "</head>"
            "<body>"
            "<h1>%s</h1>"
            "<h3>%s</h3>"
            "<h1>%s</h1>"
            "<br><br>"
            "<h3>Other Translations</h3>"
            "<h1>%s</h1>"
            "</body>"
            "</html>" % (word.title(), self.dico[word][0], trnslatn[0], trnslatn[1]))

```

```
def loadDataBase(self):
```

```

    content = list(self.dico.keys())
    self.dataBaseList.addItem(content)

```

```
def showMessage(self, title, msg, type):
```

```
    """
```

```

    :param title: the title of the msh
    :param msg: the content msg
    :param type: the type of message, e.g error,information
    :return: nothing
    """

```

```
    if type.lower() == "information":
```

```
        QtWidgets.QMessageBox.information(QtWidgets.QMessageBox(),title,msg)
```

```
    else:
```

```
        QtWidgets.QMessageBox.warning(QtWidgets.QMessageBox(),title,msg)
```

```
def database(self):
```

```
    if self.databaseContainer.isVisible():
```

```
        self.databaseContainer.setVisible(False)
```

```
        self.actionWord_DataBase.setText("Show database")
```

```

else:
    self.databaseContainer.setVisible(True)
    self.actionWord_DataBase.setText("Hide database")

def translate(self):
    """
    This method is used to perform translation
    :return:
    """
    try:
        query = self.input_tf.text()
        status, result = smt.getMatch(query, self.dico)

        if status == True: #the exact query exists in the database
            #self.output_tf.setPlainText(query.title() + " " + result[0] + "\n" + self.stripList(result[1]))
            translatn = self.stripList(result[1])
            if len(translatn) == 1: #for single translation
                self.output_tf.setText("<html>"
                    "<head>"
                    "</head>"
                    "<body>"
                    "<h1>%s</h1>"
                    "<h3>%s</h3>"
                    "<h1>%s</h1>"
                    "</body>"
                    "</html>" % (query.title(), result[0], translatn[0]))

            else:#when there is more than one translation
                self.output_tf.setText("<html>"
                    "<head>"
                    "</head>"
                    "<body>"
                    "<h1>%s</h1>"
                    "<h3>%s</h3>"
                    "<h1>%s</h1>"
                    "<br><br>"
                    "<h3>Other Translations</h3>"
                    "<h1>%s</h1>"
                    "</body>"
                    "</html>" % (query.title(), result[0], translatn[0], translatn[1]))

        self.stackedWidget.setVisible(False) #hide the suggestion panel
    else:
        if len(result) > 0: #if any similar words was found

```

```

        self.matches = result[:]
        print(self.matches)
        self.lastword = self.matches.pop().word
        self.suggestion_tf.setPlainText("Do you mean: "+self.lastword+"?")
        self.stackedWidget.setCurrentIndex(0)
        self.stackedWidget.setVisible(True)
    else:
        self.showMessage("Translator", "Word not found!!", "information")

except Exception as e:
    self.showMessage("Error", str(e), "warning")

def yesAction(self):
    """
    This method is called when the yes button is pressed
    :return:
    """
    #self.output_tf.setPlainText(self.lastword.title() + " " + self.dico[self.lastword][0] + "\n" +
self.stripList( self.dico[self.lastword][1] ))
    trnsln = self.stripList( self.dico[self.lastword][1])
    if len(trnsln) == 1:
        self.output_tf.setText("<html>"
            "<head>"
            "</head>"
            "<body>"
            "<h1>%s</h1>"
            "<h3>%s</h3>"
            "<h1>%s</h1>"
            "</body>"
            "</html>" % (self.lastword.title(), self.dico[self.lastword][0], trnsln[0] ) )
    else:
        self.output_tf.setText("<html>"
            "<head>"
            "</head>"
            "<body>"
            "<h1>%s</h1>"
            "<h3>%s</h3>"
            "<h1>%s</h1>"
            "<br><br>"
            "<h3>Other Translations</h3>"
            "<h1>%s</h1>"
            "</body>"
            "</html>" % ( self.lastword.title(), self.dico[self.lastword][0], trnsln[0], trnsln[1]
))

```

```
self.stackedWidget.setVisible(False)
self.suggestion_tf.setPlainText("")
```

```
def noAction(self):
    if len(self.matches) > 0:
        self.lastword = self.matches.pop().word
        self.suggestion_tf.setPlainText("Do you mean: " + self.lastword + "?")
    else:
        self.showMessage("Notification", "No match found!", "information")
        self.stackedWidget.setVisible(False)
```

```
def stackAction(self):
    if self.stackedWidget.currentIndex() == 0:
        self.stackedWidget.setCurrentIndex(1)
    else:
        self.stackedWidget.setCurrentIndex(0)
```

```
def stripList(self,dlist):
    """
    splits the strings in the list if more than one into two, the first word(fword) and the remaining
    words(rword)
    :param dlist: the list whose words will be concatenated into a single string
    :return: the only fword if there is only one word in the list, or returns the fword and rword.
    """
    if len(dlist) > 0:
        fword = dlist[0]
        if len(dlist) > 1: #if there are more than one meaning
            rword = ", ".join(dlist[1:])
            return (fword, rword)
        else:
            return (fword,)
```

```
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "CPE/13/1070"))
    self.titleLabel.setText(_translate("MainWindow", "ENGLISH - YORUBA WORD TRANSLATOR"))
    self.input_tf.setPlaceholderText(_translate("MainWindow", "Enter text here"))
    self.translateButton.setText(_translate("MainWindow", "Translate"))
    self.yesButton.setText(_translate("MainWindow", "Yes"))
```

```
self.noButton.setText(_translate("MainWindow", "No"))
self.display_groupBox.setTitle(_translate("MainWindow", "Display"))
self.databaseContainer.setTitle(_translate("MainWindow", "Word Datas"))
self.menuABOUT.setTitle(_translate("MainWindow", "Help"))
self.actionAbout.setText(_translate("MainWindow", "About"))
self.actionAbout.setShortcut(_translate("MainWindow", "Ctrl+H"))

self.actionWord_DataBase.setText(_translate("MainWindow", "Show DataBase"))
self.actionWord_DataBase.setShortcut(_translate("MainWindow", "Ctrl+D"))
self.actionExit.setText(_translate("MainWindow", "Exit"))
self.actionExit.setShortcut(_translate("MainWindow", "Alt+F4"))
#hide the suggestion widget and database list
self.stackedWidget.setVisible(False)
self.databaseContainer.setVisible(False)
```

```
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = MyWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```